

FINITE ELEMENT SOLUTION OF THE NONLINEAR
COUPLED NEUTRONIC-ENERGY EQUATIONS
FOR A FAST REACTOR FUEL CELL

Roy Edward Kasdorf

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

FINITE ELEMENT SOLUTION OF THE NONLINEAR
COUPLED NEUTRONIC-ENERGY EQUATIONS
FOR A FAST REACTOR FUEL CELL

by

Roy Edward Kasdorf

December 1976

Thesis Advisor:
Thesis Advisor:

D. Nguyen
D. Salinas

Approved for public release; distribution unlimited.

T176641

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Finite Element Solution of the Non-linear Coupled Neutronic-Energy Equations for a Fast Reactor Fuel Cell		5. TYPE OF REPORT & PERIOD COVERED Master's & Mechanical Engineer's Thesis; December 1976
7. AUTHOR(s) Roy Edward Kasdorf		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		12. REPORT DATE December 1976
		13. NUMBER OF PAGES 155
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A transient overpower (TOP) accident in a Liquid Metal Fast Breeder Reactor (LMFBR) is considered. The analysis is formulated to model the dynamic response of the reactor fuel subassembly during the initial period of the postulated overpower transient. An equivalent cylindrical cell is used to model the fuel subassembly. The governing neutronic and heat transport equations for each region (fuel, clad, and		

coolant) of the equivalent cylindrical cell are developed. Nuclear Doppler broadening feedback is included in the dynamic model making the coupled equations non-linear. The resulting non-linear partial differential field equations are transformed into a system of ordinary differential equations by the finite element method. An isoparametric, quadratic, rectangular element is used for the discretization of the spatial domain. When using the finite element method, large system matrices may result. To facilitate solution of these large systems, an optimum compacting scheme is utilized. The implicit Gear's method is used for the solution of the system of ordinary differential equations. The results for a sample problem are presented.

Finite Element Solution of the Nonlinear
Coupled Neutronic-Energy Equations
for a Fast Reactor Fuel Cell

by

Roy Edward Kasdorf
Lieutenant, United States Navy
B.S.M.E., New Mexico State University, 1970

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

and

MECHANICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
December 1976

ABSTRACT

A transient overpower (TOP) accident in a Liquid Metal Fast Breeder Reactor (LMFBR) is considered. The analysis is formulated to model the dynamic response of the reactor fuel subassembly during the initial period of the postulated overpower transient. An equivalent cylindrical cell is used to model the fuel subassembly. The governing neutronic and heat transport equations for each region (fuel, clad, and coolant) of the equivalent cylindrical cell are developed. Nuclear Doppler broadening feedback is included in the dynamic model making the coupled equations non-linear. The resulting non-linear partial differential field equations are transformed into a system of ordinary differential equations by the finite element method. An isoparametric, quadratic, rectangular element is used for the discretization of the spatial domain. When using the finite element method, large system matrices may result. To facilitate solution of these large systems, an optimum compacting scheme is utilized. The implicit Gear's method is used for the solution of the system of ordinary differential equations. The results for a sample problem are presented.

TABLE OF CONTENTS

I.	INTRODUCTION - - - - -	13
II.	DESCRIPTION OF PROBLEM - - - - -	16
	A. PHYSICAL SYSTEM - - - - -	16
	B. SYSTEM MODEL - - - - -	18
	C. NUMERICAL SOLUTION - - - - -	20
III.	MODEL DEVELOPMENT - - - - -	22
	A. NEUTRONIC ANALYSIS - - - - -	22
	1. Fuel Region - - - - -	23
	2. Cladding Region - - - - -	25
	3. Coolant Region - - - - -	26
	4. Infinite Multiplication Factor - - - - -	26
	5. Boundary Conditions - - - - -	30
	B. HEAT TRANSFER ANALYSIS - - - - -	32
	1. Fuel Region - - - - -	32
	2. Cladding Region - - - - -	33
	3. Coolant Region - - - - -	33
	4. Interface Conditions - - - - -	34
	5. Boundary Conditions - - - - -	36
IV.	FINITE ELEMENT FORMULATION - - - - -	39
	A. BASIC THEORY - - - - -	39
	B. SHAPE FUNCTIONS - - - - -	42
	C. COORDINATE TRANSFORMATIONS - - - - -	43
V.	APPLICATION OF FEM TO GOVERNING FIELD EQUATIONS - - - - -	48
	A. GAUSSIAN QUADRATURE - - - - -	48

B.	NEUTRONIC FIELD EQUATIONS	- - - - -	49
1.	Fuel Region	- - - - -	49
2.	Clad Region	- - - - -	53
3.	Coolant Region	- - - - -	53
C.	HEAT TRANSPORT FIELD EQUATIONS	- - - - -	54
1.	Fuel Region	- - - - -	54
2.	Clad Region	- - - - -	57
3.	Coolant Region	- - - - -	58
D.	DISCRETIZATION OF THE SPATIAL DOMAIN	- - - - -	60
E.	OPTIMUM COMPACTING SCHEME	- - - - -	63
VI.	NUMERICAL SOLUTION	- - - - -	66
A.	SELECTION OF METHOD	- - - - -	66
B.	USER SUPPLIED SUBROUTINES TO IMPLEMENT THE IMPLICIT GEAR'S METHOD	- - - - -	67
1.	DIFFUN	- - - - -	67
2.	JACMAT	- - - - -	68
3.	NUITSL	- - - - -	69
VII.	PROCEDURE	- - - - -	71
A.	INPUT DATA	- - - - -	72
VIII.	RESULTS	- - - - -	75
IX.	RECOMMENDATIONS	- - - - -	87
APPENDIX A:	DEVELOPMENT OF TRANSFORMATIONS	- - - - -	89
APPENDIX B:	REDUCTION OF SECOND ORDER TERM	- - - - -	92
APPENDIX C:	LIST OF RELATIONS FOR MATERIAL THERMAL PROPERTIES	- - - - -	93
COMPUTER PROGRAM	- - - - -	- - - - -	95
LIST OF REFERENCES	- - - - -	- - - - -	153
INITIAL DISTRIBUTION LIST	- - - - -	- - - - -	155

LIST OF TABLES

I. Physical Parameters - - - - -	73
----------------------------------	----

LIST OF FIGURES

1.	Equivalent Cylindrical Cell	- - - - -	17
2.	Doppler Broadening of a Resonance Peak	- - - - -	28
3.	Gap Heat Transfer Coefficient	- - - - -	35
4.	Element Transformation	- - - - -	44
5.	Normalized Shape Functions	- - - - -	45
6.	Finite Element Discretization	- - - - -	62
7.	Sample Data Deck	- - - - -	74
8.	Convergence of the Finite Element Method	- - - - -	76
9.	Determination of the Critical Fission Cross Section	- - - - -	77
10.	Flux Profile ($\rho = \$10$)	- - - - -	79
11.	Temperature Profile ($\rho = \$10$)	- - - - -	80
12.	Radial Flux ($\rho = \$10$)	- - - - -	82
13.	Axial Flux ($\rho = \$10$)	- - - - -	83
14.	Radial Temperature Profile ($\rho = \$10$)	- - - - -	84
15.	Axial Temperature Profile ($\rho = \$10$)	- - - - -	85

LIST OF SYMBOLS AND NOTATION

A. NOTATION

$\langle \quad \rangle$	Row vector
$\{ \quad \}$	Column vector
$[\quad]$	Square matrix or indicates a reference
$[\quad]^{-1}$	Inverse of a square matrix
∇	Del operator
$\frac{\partial}{\partial x}$	Partial derivative with respect to x
Δx	Change in x
\int_v	Volume integral
\int_x	Integration with respect to x
$\det [x]$	Determinant of x

B. SYMBOLS

b	Nuclear Doppler constant
C	Concentration of delayed neutron precursors
C_p	Specific heat [cal/gm °C]
$D(\underline{r})$	Neutron diffusion coefficient [cm]
e	Nuclear energy released per fission [cal/fission]
h	Heat transfer coefficient [cal/cm ² sec °C]
J	Jacobian matrix
$J(\underline{r}, t)$	Neutron current [$\frac{\text{neutrons}}{\text{cm}^2 \text{ sec}}$]
k_∞	Infinite multiplication factor
K_D	Doppler constant
$k(\underline{r})$	Thermal conductivity [cal/cm sec °C]
LMFBR	Liquid Metal Fast Breeder Reactor

LOCA	Loss of Coolant Accident	
N	Shape function	
n	Number of delay neutron groups	
$n(\underline{r},t)$	Neutron density	$[\frac{\text{neutrons}}{\text{cm}^3}]$
N,x	Derivative of N with respect to x	
$\dot{q}(\underline{r},t)$	Nuclear generation	$[\text{cal}/\text{cm}^3\text{sec}]$
R	Residual	
\underline{r}	Spatial coordinate	$[\text{cm}]$
r,z	Global coordinates	
$S(\underline{r},t)$	Neutron production	$[\frac{\text{neutrons}}{\text{cm}^3\text{sec}}]$
T	Temperature	$[^\circ\text{C}]$
t	Time	$[\text{sec}]$
TOP	Transient Overpower	
v	Neutron velocity	$[\text{cm}/\text{sec}]$
V_{co}	Velocity of coolant flow	$[\text{cm}/\text{sec}]$
W	Weighting function	
β	Fraction of fission neutrons which appear as delayed neutrons	
$\phi(\underline{r},t)$	Neutron flux	$[\frac{\text{neutrons}}{\text{cm}^2\text{sec}}]$
λ	Decay constant of the delayed neutron precursors	$[1/\text{sec}]$
η,ξ	Local coordinates	
ν	Average number of neutrons released per fission	
ρ	Reactivity	
$\rho(\underline{r})$	Density	$[\text{gm}/\text{cm}^2]$
Σ	Neutron cross section	$[\text{cm}^{-1}]$

C. SUBSCRIPTS

a	Absorption
c	Clad
co	Coolant
CR	Critical
D	Delayed, Doppler
f	Fission
F	Fuel
gap	Fuel-clad interface
i,j	Group, equation
P	Prompt
surf	Clad-coolant interface

D. SUPERSCRIPITS

e	Element
e*	Adjacent element
o	At time zero
.	Derivative with respect to time

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to Dr. Dong Nguyen and to Dr. David Salinas, Professors of Mechanical Engineering, for their advice and guidance throughout the course of this work.

The author is obligated to Dr. Richard Franke, Professor of Mathematics, for his assistance in implementing the integration procedure and for his assistance as second reader.

The author wishes to thank Dr. Gilles Cantin, Professor of Mechanical Engineering, for his invaluable discussions concerning the finite element method.

Finally, the author wishes to thank his wife, Gail, for her encouragement and understanding throughout the course of this study.

I. INTRODUCTION

As the world's fossil fuel resources are depleted, more emphasis is being placed on the breeder reactor as a potential means of solving the coming energy crisis. While the development of new energy sources is being pushed, equal effort is being given to the maintenance of an environmentally clean world. To this end, the safety of breeder reactors is receiving a considerable amount of attention before assuming that the breeder reactor is the answer to the energy problem.

The Liquid Metal Fast Breeder Reactor (LMFBR) appears to be one of the most promising breeder reactors. Most engineers will concede there is little probability of a nuclear explosion occurring in the operation of a nuclear reactor. Of major concern to engineers is the loss of coolant accident (LOCA) and the transient overpower accident (TOP). The present analysis is concerned with a TOP accident in a LMFBR. The analysis is formulated to model the dynamic response of the reactor fuel subassembly during the initial period of the postulated overpower transient. The primary consideration is given to the early response of this fuel subassembly to various conditions of disturbances. The phenomenon which occurs after core disassembly (i.e., clad melting) is not the concern of this analysis. Only the time prior to clad melting is being considered.

No consideration is given here as to how the overpower transient occurs or to why the safety features of the reactor did not operate properly. It is postulated that the accident has occurred. In this analysis, the TOP accident is created by either a step increase in reactivity, a ramp increase in reactivity, or a combination of both.

An inherent safety feature of most reactors, nuclear Doppler broadening feedback, is included in the dynamic model of the fuel subassembly. The Doppler feedback acts to reduce the effect of the excursion. Consideration of this feedback creates a non-linear system model which is described by a non-linear, initial-boundary-value problem.

The conventional method of solution uses the standard point kinetics formulation. Recent studies have pointed out a non-negligible error in this model [1], particularly with asymmetric disturbances [2], or space-dependent feedback [3]. In Ref. [4], a somewhat novel approach of using the finite element method (FEM) for the space-time dependent solution of the reactor dynamics problem was demonstrated. The FEM is effective in handling these asymmetric disturbances and space-dependent feedbacks. Therefore, the finite element method was used so that the spatial effects on the postulated problem may be studied further.

The purpose of this work was to demonstrate further the applicability of the FEM to the non-linear reactor dynamics problem as well as to investigate the dynamic response of the reactor fuel subassembly. The analysis required a novel

approach to handle the gap conductances present at the interfaces of the equivalent cell model of the fuel subassembly; this will be clarified in the analysis.

II. DESCRIPTION OF PROBLEM

A. PHYSICAL SYSTEM

The typical Liquid Metal Fast Breeder Reactor (LMFBR) core consists of many hexagonal modules, each containing several hundred fuel pins. For this analysis, an equivalent cylindrical cell is used to model the fuel subassembly; see Figure 1. The use of equivalent cells as models for larger systems has been common practice in nuclear analysis (i.e., the well known Wigner-Sietz method). In using an equivalent cell, the actual shape of the reactor core is not important, and the analysis is applicable to any reactor which has the same equivalent cell.

The equivalent cell considered in this analysis, Figure 1, is fueled with enriched uranium dioxide, has a stainless steel cladding, and has liquid sodium for a coolant. The dimensions used are

$$a = 0.254 \text{ cm,}$$

$$b = 0.292 \text{ cm,}$$

$$c = 0.365 \text{ cm,}$$

and $H = 33.0 \text{ cm.}$

The gap between the fuel and cladding is very small and, in fact, may be nonexistent as in bonded fuels. The dimension of this gap has been assumed negligible. The height, H , of the fuel rod is shorter than many proposed systems (Fast Flux Testing Facility and Clinch River Breeder Reactor). However,

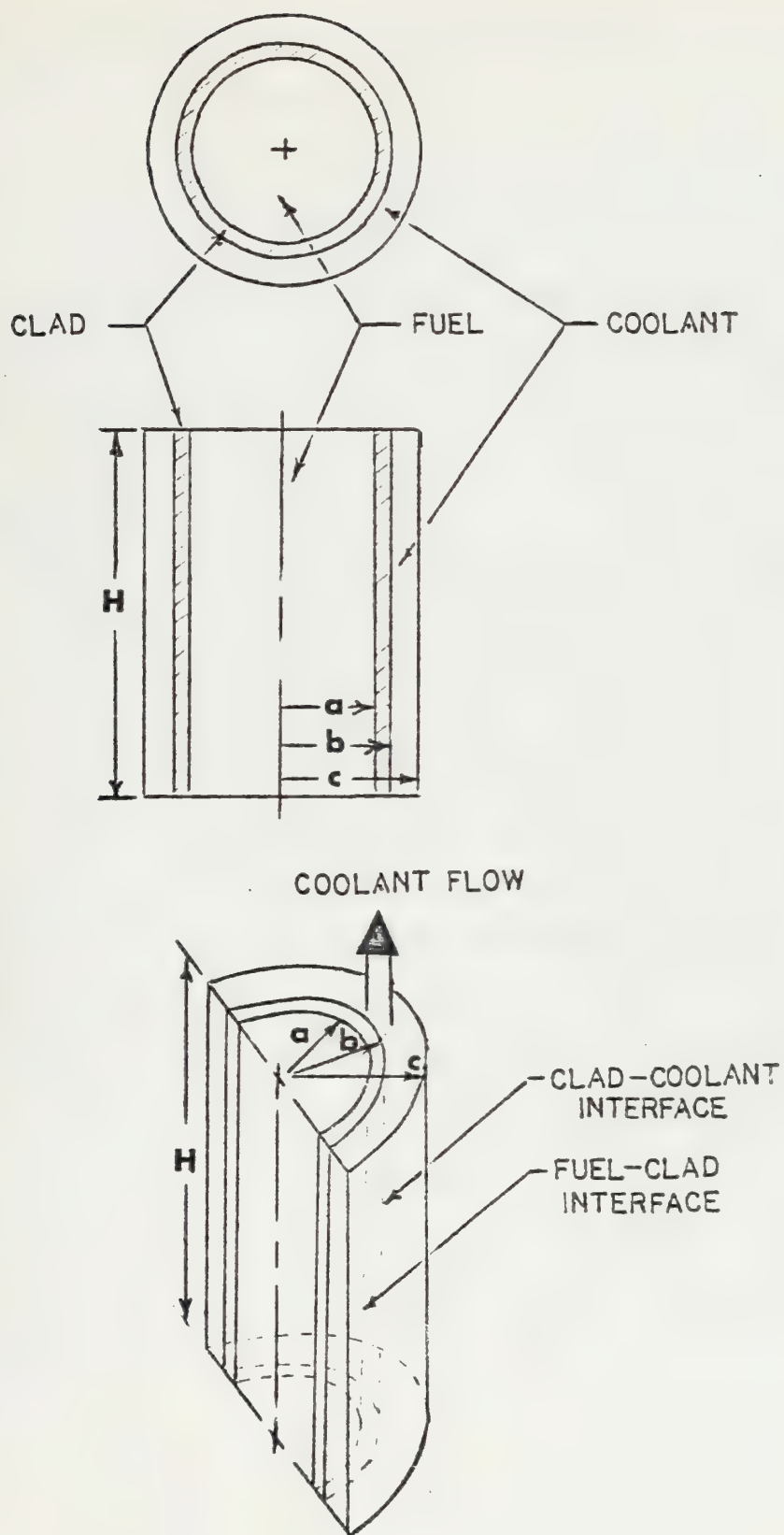


Figure 1. Equivalent Cylindrical Cell

to facilitate the numerical solution a smaller rod was used. The dynamic behavior prior to fuel pin failure for this system should be similar to the behavior of larger systems.

The treatment of this problem in three dimensions would be prohibitive in computer usage. Therefore, azimuthal symmetry is assumed, and the problem becomes a two-dimensional cylindrical (r,z) problem.

B. SYSTEM MODEL

The analysis considers the monoenergetic neutron diffusion approximation to model the transient neutron transport problem. A simple conduction-convection heat transfer model is used for the energy transport problem. The temperatures in the model are directly coupled to the neutron flux through the nuclear heat generation within the fuel. The neutron population is in turn coupled to the temperature through any of a number of reactivity feedback mechanisms. The nuclear Doppler effect is perhaps the most important of these mechanisms since it provides a negative temperature coefficient which increases the inherent stability of the reactor. Prior to core disassembly and fuel melting, the nuclear Doppler effect is the most dominant feedback and is, therefore, the only feedback mechanism considered in this analysis.

For irradiated, mixed-oxide fuels, a phenomenon of fuel restructuring has been commonly observed. This restructuring, essentially a change of phase of the fuel material, presents a unique heat transfer problem particularly during transient conditions. The problem has not been fully characterized and

is beyond the scope of this analysis. The fuel was, therefore, assumed to be a homogeneous mixture of enriched uranium dioxide.

At the fuel-cladding interface, there exists a gap which produces a thermal resistance. This thermal resistance is one of the most significant deterrents to the energy transfer to the coolant. The interface may be in physical contact or an actual gap may exist. The prediction of the thermal resistance is extremely complicated and must take into consideration many parameters: initial dimensions, type of bond, fill gas composition, fuel restructuring, fuel swelling, prior fuel life cycle, to mention a few. Reference [5] documents a computer program which attempts to predict the gap conductance, h_{gap} . This treats the thermal resistance at the interface in the same manner as a convection heat transfer coefficient when considering convection heat transfer. Since it is not the objective of this analysis to predict the gap coefficient, a representative set of values for gap coefficient, as given in Ref [6], is used in this analysis. These values are assumed to remain static during the transient. The gap conductance profile actually varies with time and will have an effect upon the transient, as noted in Ref. [7]. The prediction of this variance was not considered important for this analysis; therefore, the static assumption was made.

An average convection heat transfer coefficient was used to determine the heat transfer from the cladding to the coolant. The value used was determined from an empirical formula given in Ref. [5] and repeated in Appendix C.

In this work, consideration is given to step and ramp increases in reactivity, although any reactivity transient may easily be considered. The step and ramp increases in reactivity probably represent the most realistic physical reactivity inputs in a reactor. Once the reactivity has been inserted, the transient overpower excursion begins. Unless the Doppler feedback can override the inserted reactivity, the excursion will continue until there is physical core disassembly.

C. NUMERICAL SOLUTION

The system of equations which models the proposed problem is a non-linear, initial-boundary-value problem. The conventional method of solution of the reactor dynamics is the point kinetics formulation. It was pointed out in Refs. [1], [2], [3], and [4], that there is a non-negligible error in this model, particularly under conditions of asymmetric disturbances or space-dependent feedback. Reference [4] demonstrates the somewhat novel approach of using the finite element method (FEM) to solve the space-time dependent reactor dynamics problem. As shown in Ref. [4], the FEM is quite effective in handling localized perturbations and space-dependent feedback. In this work, only uniform disturbances were considered; however, the feedback model was space-dependent. Therefore, the finite element method is used to solve the non-linear, coupled, space-time dependent neutronic and heat transport field equations. The solution technique results in a large computer storage requirement; therefore, an optimum compact storage scheme, Ref. [8], is utilized for storage of the discretized matrices.

Once the domain has been discretized by the FEM, the solution of the resulting ordinary differential equations was to be accomplished by the implicit Gear's method, Ref. [9].

III. MODEL DEVELOPMENT

A. NEUTRONIC ANALYSIS

In this section the governing field equations for the neutron population (flux) for each of the three regions (fuel, clad, and coolant) of the domain will be formulated. The monoenergetic, diffusion theory will be used.

Consider an arbitrary volume of material within a reactor. Applying the condition of conservation to the monoenergetic neutrons leads to the neutron equation of continuity [10]

$$\frac{\partial n(\underline{r},t)}{\partial t} = S(\underline{r},t) - \Sigma_a(\underline{r})\phi(\underline{r},t) - \text{div } J(\underline{r},t) \quad (1)$$

where

\underline{r} - spatial point

t - time

$n(\underline{r},t)$ - neutron density

$S(\underline{r},t)$ - neutron production

$\Sigma_a(\underline{r})$ - neutron absorption cross section

$\phi(\underline{r},t)$ - neutron flux

$J(\underline{r},t)$ - neutron current

The left-hand side of equation (1) represents the time rate of change of the neutron density which is related to flux, ϕ , by

$$n(\underline{r},t) = \frac{1}{V} \phi(\underline{r},t) \quad (2)$$

where

v - neutron velocity

On the right-hand side of equation (1), the first term is neutron production, the second term is a neutron loss through absorption, and the third term is a neutron loss through leakage from the control volume.

Using equation (2) and applying Fick's Law to the equation of continuity results in the classical neutron diffusion equation

$$\nabla \cdot (D(\underline{r}) \nabla \phi(\underline{r}, t)) - \Sigma_a(\underline{r}) \phi(\underline{r}, t) + S(\underline{r}, t) = \frac{1}{v} \frac{\partial \phi(\underline{r}, t)}{\partial t} \quad (3)$$

where $D(\underline{r})$ - neutron diffusion coefficient

The vector notation used here is intended to include only two dimensions (r, z) since azimuthal symmetry has been assumed.

Equation (3) is applicable to each of the three regions of the equivalent cylindrical cell. The subscripts F (fuel), c (cladding), and co (coolant), will be used to denote these regions.

1. Fuel Region

In applying equation (3) to the fuel region the material properties of the fuel must be used. Within the fuel the neutron source term is due to the nuclear fission process. During fission, neutrons are released as both prompt neutrons and delayed neutrons so that

$$S(\underline{r}, t) = S_p(\underline{r}, t) + S_D(\underline{r}, t) \quad (4)$$

where $S_p(\underline{r}, t)$ - prompt neutron source
 $S_D(\underline{r}, t)$ - delayed neutron source

The neutron sources are commonly represented as [10]

$$S_p = k_{\infty} \Sigma_{aF} \phi_F (1 - \beta) \quad (5)$$

and

$$S_D = \sum_{i=1}^n C_i \lambda_i \quad (6)$$

where

k_{∞} - infinite multiplication factor

β - fraction of fission neutrons which appear as delayed neutrons

n - number of delayed neutron groups

C_i - concentration of delayed neutron precursors in the i th group

λ_i - decay constant of the delayed neutrons

The space and time variables, \underline{r} and t , will be dropped except where needed for clarification.

The concentration of delayed neutron precursors, C_i , is given by the following first order partial differential equation [10]

$$\frac{\partial C_i}{\partial t} = \beta_i k_{\infty} \Sigma_{aF} \phi_F - \lambda_i C_i \quad (7)$$

where

β_i - fraction of delayed neutrons which appear as delayed neutrons in the i th group



The solution of equation (7) is

$$C_i = \beta_i \int_0^t e^{-\lambda_i(t-t')} k_\infty(r, t') \Sigma_{aF} \phi_F dt' + C_i^0 e^{-\lambda_i t} \quad (8)$$

C_i^0 - concentration of delayed neutron precursors of the ith group at time zero

Reference [10] develops an expression for the initial concentration of delayed neutrons

$$C_i^0 = \beta_i k_\infty^0 \Sigma_{aF} \phi^0 / \lambda_i \quad (9)$$

where

k_∞^0 - initial finite multiplication factor

ϕ^0 - initial neutron flux

Combining equations (3), (4), (5), (6), (8), and (9), yields the governing equation for the fuel region

$$\begin{aligned} & \nabla \cdot (D_F \nabla \phi_F) + \Sigma_{aF} \phi_F [k_\infty(1-\beta) - 1] \\ & + \sum_{i=1}^n \lambda_i [\beta_i \int_0^t e^{-\lambda_i(t-t')} k_\infty(\underline{r}, t') \phi_F \Sigma_{aF} dt' \\ & + \beta_i k_\infty^0 \Sigma_{aF} \phi_F / \lambda_i e^{-\lambda_i t}] = \frac{1}{v} \frac{\partial \phi_F}{\partial t} \end{aligned} \quad (10)$$

2. Cladding Region

The cladding separates the fuel and coolant and contains the fuel and the fission by-products. Equation (3) governs the neutron flux in the clad. Since the clad contains

no fissile material, the neutron source term is zero. The field equation is, then,

$$\nabla \cdot (D_c \nabla \phi_c) - \Sigma_{ac} \phi_c = \frac{1}{v} \frac{\partial \phi_c}{\partial t} \quad (11)$$

3. Coolant Region

The annular region around the cladding of the equivalent cell contains the coolant. As in the clad, there is no fissile material in the coolant and, therefore, no neutron source term. The equation governing the neutron flux in the coolant is

$$\nabla \cdot (D_{co} \nabla \phi_{co}) - \Sigma_{aco} \phi_{co} = \frac{1}{v} \frac{\partial \phi_{co}}{\partial t} \quad (12)$$

Equations (10), (11), and (12) are the one-velocity, diffusion approximation used to model the neutron transport problem.

4. Infinite Multiplication Factor

The infinite multiplication factor, k_∞ , may be expressed as the infinite multiplication factor at time zero (start of transient), k_∞^0 , plus the postulated reactivity insertion (such as a step or a ramp), ρ , minus the change in the Doppler reactivity feedback, $\Delta \rho_D$. Other feedback mechanisms are normally not as significant as the Doppler broadening feedback prior to fuel melting and have been neglected in this analysis. Therefore,

$$k_\infty = k_\infty^0 + \rho - \Delta \rho_D \quad (13)$$

For a fast reactor, k_{∞}° may be approximated as

$$k_{\infty}^{\circ} = \nu \frac{\Sigma_{fF}}{\Sigma_{aF}} \quad (14)$$

where

ν - average number of neutrons released per fission

Σ_{fF} - fission cross section of the fuel

Σ_{aF} - absorption cross section of the fuel

The nuclear Doppler effect is a very important safety feature in a nuclear reactor. Nuclei in an atom are in continual motion due to their own thermal energy. As a result of this motion, even when monoenergetic neutrons interact with the atom, there appears to be a spread in the energy of the neutron - the Doppler effect. It can be shown that the cross section of a resonance becomes less in magnitude and wider as the motion of the nuclei increases [10]. As the temperature increases, the motion increases, and the shape of a resonance cross section broadens. This broadening increases the average cross section, thus, providing a negative temperature coefficient. This effect is shown in Figure 2. It is this nuclear Doppler broadening effect which provides one of the few inherent, reliable, negative reactivity feedbacks which slows an overpower transient and possibly stops a mild overpower transient.

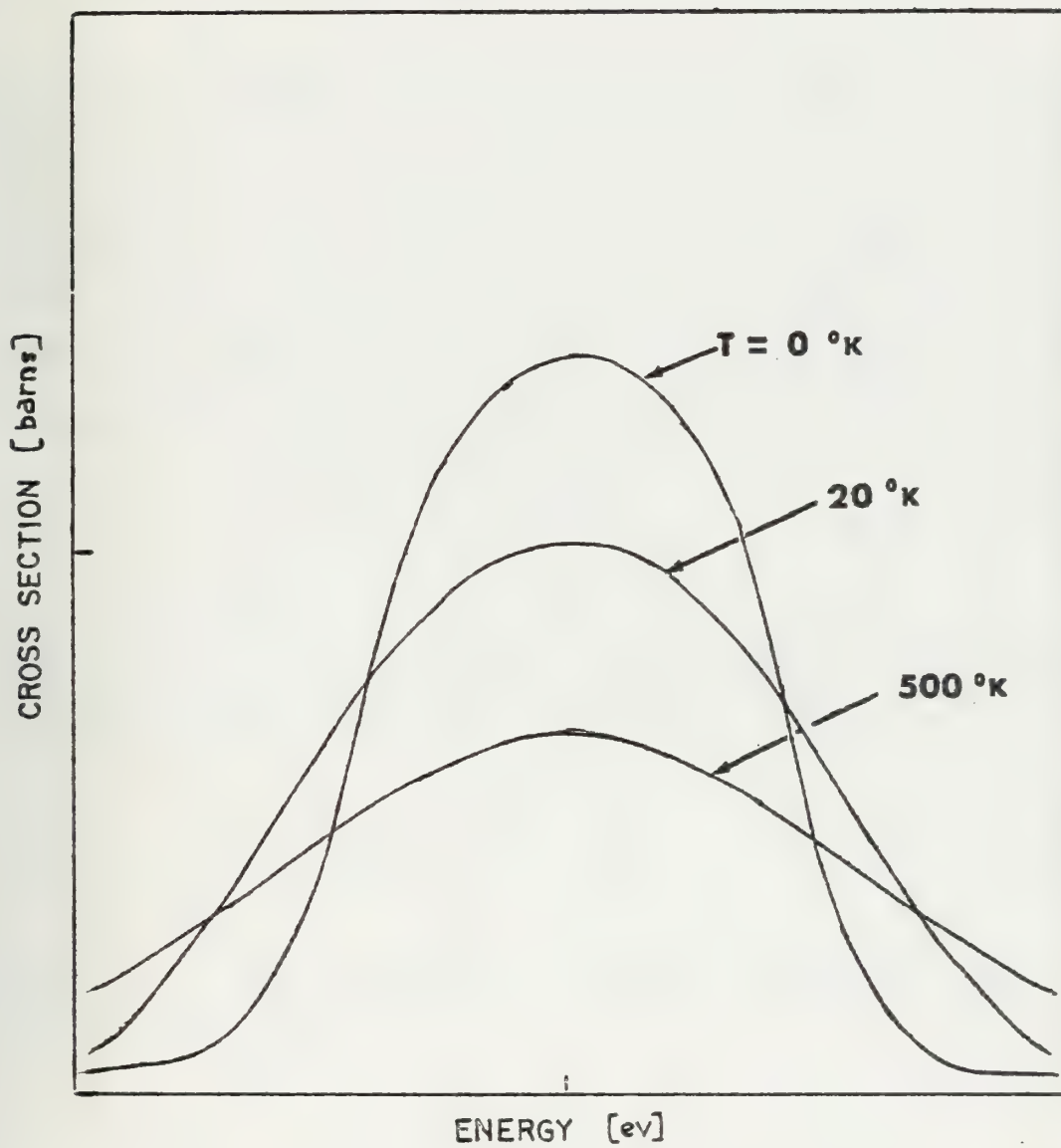


Figure 2. Doppler Broadening of a Resonance Peak

The Doppler reactivity change with respect to fuel temperature changes should be written as [6]

$$\frac{d\rho_D}{dT} = aT^{-3/2} + bT^{-1} + cT^{m-1} \quad (15)$$

where a , b , and c are parameters determined from experimental work and m is an integer. However, as noted in Ref. [6], a substantial amount of work has shown that $T \frac{d\rho_D}{dT}$ is very nearly constant over the temperature range under consideration. Therefore, the coefficients a and c have been set equal to zero, and b is defined as

$$b = K_D = T \frac{d\rho_D}{dT} \quad (16)$$

The constant, K_D , is commonly called the Doppler constant. Solving equation (16) for ρ_D yields

$$\rho_D = b \ln T_F + K \quad (17)$$

where K is an integration which may be obtained from initial conditions.

$$K = \rho_D^0 - b \ln T_F^0 \quad (18)$$

where

ρ_D^0 - Doppler effect at time zero

T_F^0 - fuel temperature at time zero

Substituting for K in equation (17) will give

$$\rho_D - \rho_D^0 = \Delta\rho_D = b \ln(T_F/T_F^0) \quad (19)$$

The infinite multiplication factor now becomes

$$k_\infty = k_\infty^0 + \rho - b \ln(T_F/T_F^0) \quad (20)$$

The effect of delayed neutrons is small compared to the prompt neutron effect; therefore, it may be assumed that the Doppler effect on delayed neutrons is insignificant to the overall problem. The k_∞ of equation (8) is, then, assumed to be k_∞^0 . With this assumption and equation (20), the neutron diffusion equation in the fuel, equation (10), may be rewritten as

$$\begin{aligned} \nabla \cdot (D_F \nabla \phi_F) + \Sigma_{aF} \phi_F [k_\infty^0(1-\beta) - 1 + \rho(1-\beta) - b(1-\beta) \ln(T_F/T_F^0)] \\ + \sum_{i=1}^n \lambda_i \{ \beta_i \Sigma_{aF} \int_0^t e^{-\lambda_i(t-t')} [k_\infty^0 + \rho] \phi_F dt' + C_i^0 e^{-\lambda_i t} \} = \frac{1}{v} \frac{\partial \phi_F}{\partial t} \end{aligned} \quad (21)$$

To facilitate the present analysis, the number of delayed neutron groups is taken as one averaged group. So that, λ_i and β_i become $\bar{\lambda}$ and $\bar{\beta}$, respectively. This approximation should have little effect on the problem under consideration.

5. Boundary Conditions

The neutron diffusion problem involves solution of the partial differential equations (11), (12), and (21), with the following boundary, interface, and initial conditions:

- 1) $\frac{\partial \phi_F}{\partial r} (0, z, t) = 0$
- 2) $\phi_F (a, z, t) = \phi_c (a, z, t)$
- 3) $D_F \frac{\partial \phi_F}{\partial r} (a, z, t) = D_c \frac{\partial \phi_c}{\partial r} (a, z, t)$
- 4) $\phi_c (b, z, t) = \phi_{c0} (b, z, t)$
- 5) $D_c \frac{\partial \phi_c}{\partial r} (b, z, t) = D_{c0} \frac{\partial \phi_{c0}}{\partial r} (b, z, t)$
- 6) $\frac{\partial \phi_{c0}}{\partial r} (c, z, t) = 0$
- 7) $\phi_F (r, \pm \frac{H}{2}, t) = \phi_c (r, \pm \frac{H}{2}, t) = \phi_{c0} (r, \pm \frac{H}{2}, t) = 0$
- 8) $\phi_F (\underline{r}, 0) = \phi_F^\circ (\underline{r})$
- 9) $\phi_c (\underline{r}, 0) = \phi_c^\circ (\underline{r})$
- 10) $\phi_{c0} (\underline{r}, 0) = \phi_{c0}^\circ (\underline{r})$

Boundary condition 1) results from the assumed azimuthal symmetry. Interface conditions 2), 3), 4), and 5) are continuity conditions of the flux. Boundary condition 6) results from the use of an equivalent cell and basically indicates there is an equal number of neutrons transferred in and out of the cell at the outer boundary. This should be valid unless the cell is located near the outer edge of the reactor. Boundary condition 7) is an assumption that the flux is zero at the axial boundaries of the cell. Initial conditions 8), 9), and 10) are the assumed initial distributions of the neutron flux.

B. HEAT TRANSFER ANALYSIS

In this section, the principle of conservation of energy will be used to formulate the governing field equations for the heat transport in each of the three regions. A simple heat conduction model with convection heat transfer to the coolant is used to model the heat transport problem. A gap conductance model is used to describe the heat transport across the gap at the fuel-clad interface.

1. Fuel Region

Conservation of energy within the fuel region yields the unsteady heat conduction equation with a generation term

$$\nabla \cdot (k_F(\underline{r}) \nabla T_F(\underline{r}, t)) + \dot{q}(\underline{r}, t) = \rho_F(\underline{r}) C_{pF}(\underline{r}) \frac{\partial T_F}{\partial t}(\underline{r}, t) \quad (22)$$

where

$k_F(\underline{r})$ - thermal conductivity of the fuel

$T_F(\underline{r}, t)$ - fuel temperature

$\dot{q}(\underline{r}, t)$ - nuclear energy generation per unit
volume

$\rho_F(\underline{r})$ - fuel density

$C_{pF}(\underline{r})$ - fuel specific heat

As in the neutronic analysis, the vector notation is intended to include only two dimensions, (r, z) . The \underline{r} and t will be dropped except where needed for clarification.

The nuclear generation term may be expressed as

$$\dot{q} = e \Sigma_{fF} \phi_F \quad (23)$$

where

e - nuclear energy released per fission

As can be seen, it is through the nuclear generation term that the temperature is directly coupled to the neutron flux. This coupling and the temperature dependent Doppler reactivity feedback combine to make the coupled problem non-linear.

Substituting equation (23) into equation (22) yields the governing thermal equation for the fuel

$$\nabla \cdot (k_F \nabla T_F) + e \Sigma_{fF} \phi_F = \rho_F C_{pF} \frac{\partial T_F}{\partial t} \quad (24)$$

2. Cladding Region

Conservation of energy within the clad will yield the heat conduction equation. With nuclear generation, a relatively small amount (~5%) of the energy will be released in the cladding and the coolant. However, in this analysis it is assumed that the total energy release is in the fuel region. This should not create any significant error. Using this assumption, the unsteady heat conduction equation for the cladding becomes

$$\nabla \cdot (k_C \nabla T_C) = \rho_C C_{pC} \frac{\partial T_C}{\partial t} \quad (25)$$

3. Coolant Region

Once again, conservation of energy will lead to the heat conduction equation plus an additional term which

takes into consideration the coolant flow. The governing equation is

$$\nabla \cdot (k_{co} \nabla T_{co}) - V_{co} \frac{\partial}{\partial z} (\rho_{co} C_{pco} T_{co}) = \rho_{co} C_{pco} \frac{\partial T_{co}}{\partial t} \quad (26)$$

where V_{co} - coolant flow velocity

Equations (24), (25), and (26) are the governing equations used to model the energy transport problem.

4. Interface Conditions

The interface between the fuel and the cladding may be an actual gap with a finite distance, or the surfaces may be in intermittent contact on a microscopic scale. To model the heat transfer across this interface, a gap heat transfer coefficient is introduced. The gap coefficient must take into consideration many items (e.g., radiation heat transfer across the gap, heat transfer by solid-to-solid contact, heat conduction across a gas filled gap). The prediction of this gap coefficient is extremely complicated and beyond the scope of this work. In Ref. [6], a set of values for H_{gap} is given and the axial variation of H_{gap} in this analysis is approximately the same. A cosine curve has been fitted to the sample data to determine the gap coefficient, see Figure 3. The heat flux across the fuel-clad interface is, then,

$$q = H_{gap} (T_F - T_C) \quad (27)$$

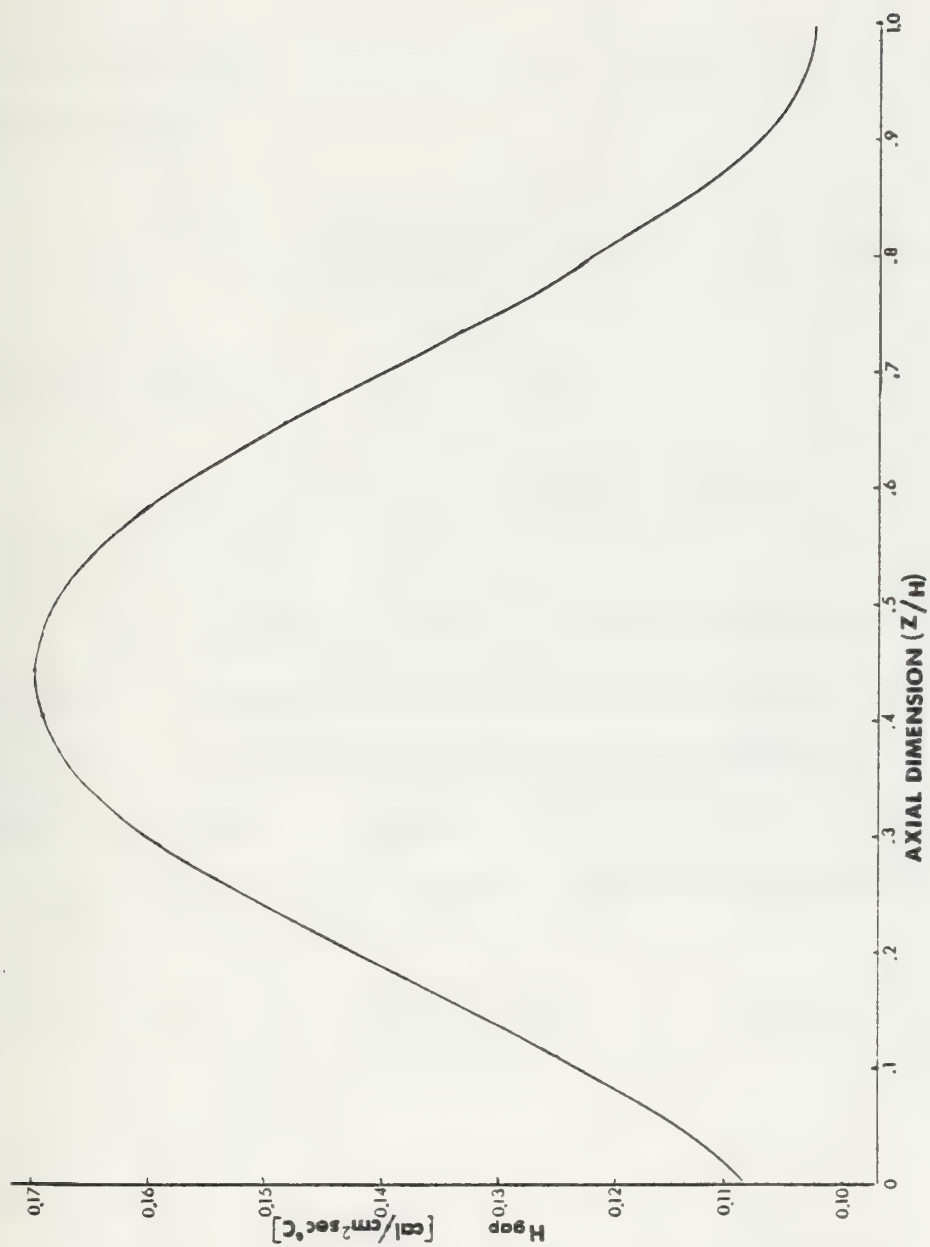


Figure 3. Gap Heat Transfer Coefficient

The heat conducted out of the fuel is governed by Fourier's equation and is equal to the heat transferred across the gap

$$q = -k_F \frac{\partial T_F}{\partial r} \quad (28)$$

Equating equations (27) and (28) gives the fuel-clad interface condition

$$T_F(a,z,t) + \frac{k_F(a,z)}{H_{\text{gap}}(z)} \frac{\partial T_F}{\partial r}(a,z,t) = T_C(a,z,t) \quad (29)$$

and from continuity

$$k_F(a,z) \frac{\partial T_F}{\partial r}(a,z,t) = k_C(a,z) \frac{\partial T_C}{\partial r}(a,z,t) \quad (29a)$$

As is common practice in convection heat transfer analysis, a coolant surface heat transfer coefficient, h_{surf} , is used to account for the thermal resistance at the clad-coolant interface. Similar to the fuel-clad interface analysis, the clad-coolant interface conditions may be determined

$$T_C(b,z,t) + \frac{k_C(b,z)}{h_{\text{surf}}} \frac{\partial T_C}{\partial r}(b,z,t) = T_{C0}(b,z,t) \quad (30)$$

and

$$k_C(b,z) \frac{\partial T_C}{\partial r}(b,z,t) = k_{C0}(b,z) \frac{\partial T_{C0}}{\partial r}(b,z,t) \quad (30a)$$

5. Boundary Conditions

The boundary and initial conditions for the heat transport problem are:

- 1) $\frac{\partial T_F}{\partial r} (0, z, t) = 0$
- 2) $T_{co}(r, -\frac{H}{2}, t) = T_{PLENUM}$
- 3) $\frac{\partial T_{co}}{\partial z} (r, \frac{H}{2}, t) = 0$
- 4) $\frac{\partial T_{co}}{\partial r} (c, z, t) = 0$
- 5) $\frac{\partial T_F}{\partial z} (r, \pm \frac{H}{2}, t) = \frac{\partial T_c}{\partial z} (r, \pm \frac{H}{2}, t) = 0$
- 6) $T_F(\underline{r}, 0) = T_F^o(\underline{r})$
- 7) $T_c(\underline{r}, 0) = T_c^o(\underline{r})$
- 8) $T_{co}(\underline{r}, 0) = T_{co}^o(\underline{r})$

Boundary condition 1) results from the assumed azimuthal symmetry. The coolant has been assumed to enter the flow channel at a constant temperature, T_{PLENUM} . This results in condition 2). Boundary conditions 3) and 5) result from an assumption that no heat is transferred axially from the fuel rod. Boundary condition 4) is the result of the use of the equivalent cell. Conditions 6), 7), and 8) are the assumed initial conditions.

Solution of the nonlinear coupled neutronic and energy transport problem involves the solution of the partial differential equations (11), (12), (21), (24), (25), and (26), with the appropriate boundary, interface, and initial conditions.

Several works, Refs. [4], [11], [12], have demonstrated the feasibility and success of the finite element method in solving nuclear reactor dynamics problems. The FEM

is used to reduce the partial differential equations developed in this analysis to a system of ordinary differential equations. Integration of these ordinary differential equations (ODE) yields the solution.

IV. FINITE ELEMENT FORMULATION

In this section, the basic theory underlying the finite element method is formulated. Selection of the finite elements and the shape functions for the elements are given. Some simple transformations which facilitate the integration necessary in the FEM are also presented.

A. BASIC THEORY

To obtain a numerical solution, the governing partial differential field equations are transformed into a system of ODE in finite dimensional vector space. This may be accomplished in several manners such as the finite-difference method, the variational method, or the weighted residual method. In this work, the Galerkin method (a weighted residual method) is utilized for the discretization of the spatial domain. The Galerkin procedure will be applied to each of the governing field equations. Any of these equations may be considered to be in the following form

$$\frac{\partial \psi}{\partial t}(\underline{r}, t) - \mathcal{L} \psi(\underline{r}, t) = f(\underline{r}, t) \quad (31)$$

where ψ represents the unknown function, e.g., in equation (21), ψ represents ϕ_F , \mathcal{L} represents the operator for each individual equation, and f is a forcing function. In the finite element method the solution is approximated as

$$\psi(\underline{r}, t) \cong \tilde{\psi}(\underline{r}, t) = \sum_{i=1}^N N_i(\underline{r}) \psi_i(t) = \langle N_i \rangle \{ \psi_i \} \quad (32)$$

where

N - the number of degrees of freedom

N_i - the element shape function

ψ_i - unknown coordinate function

$\langle \rangle$ - matrix notation for a row vector

$\{ \}$ - matrix notation for a column vector

$\langle N_i \rangle = \langle N_1 \ N_2 \ \dots \ N_i \ \dots \ N_N \rangle$

$$\{ N_i \} = \begin{Bmatrix} N_1 \\ N_2 \\ \vdots \\ N_i \\ \vdots \\ N_N \end{Bmatrix}$$

The residual, $R(\underline{r}, t)$, is a measure of the error in this finite element approximation. The residual may be considered as

$$R = \frac{\partial \psi}{\partial t} - \mathcal{L} \psi - f \quad (33)$$

The best solution for $\tilde{\psi}$ is one which "minimizes" this residual. Various "minimums" are obtained by the weighted residual method by setting

$$\int_V W_i(\underline{r}) R \, dV = 0 \quad i=1,2,\dots,N \quad (34)$$

$W_i(\underline{r})$ - weighting functions

With the Galerkin method, the weighting functions are the shape functions defining the approximation of equation (32) (i.e., $W_i = N_i$). A noteworthy attribute of the Galerkin method is the opportunity of using an integration-by-parts

of the terms involving the second order spatial derivatives. A lower order finite element may be used than would have been possible otherwise. Once the weighting functions have been chosen, the problem becomes

$$\int_V N_i \left\{ \frac{\partial \psi}{\partial t} - \mathcal{L} \psi - f \right\} dV = 0 \quad i=1,2,\dots,N \quad (35)$$

The integration involved in equation (35) is carried out on the element level, taking advantage of the use of a "local" coordinate system. Once the integration is accomplished, the results are merged into a system using "global" coordinates. On the element level

$$\tilde{\psi}^e = \langle N_j \rangle^e \{\psi_j\}^e \quad j=1,2,\dots,N \quad (36)$$

where the superscript e indicates the element level. Substituting $\tilde{\psi}^e$ into equation (35) and noting $\{\psi_j\}^e$ is not a function of the spatial domain yields

$$\int_V \langle N_i \rangle^e \{N_j\}^e dV \{\dot{\psi}_j\}^e - \int_V [\langle N_i \rangle^e \mathcal{L} \{N_j\}^e - \langle N_i \rangle^e f^e] dV \{\psi_j\}^e = 0 \quad (37)$$

where $i, j = 1, 2, \dots, N^e$
 N^e - number of degrees of freedom for an element

The operator \mathcal{L} will vary depending upon which governing equation is under consideration.

B. SHAPE FUNCTIONS

The shape functions, N_i , are chosen to satisfy certain completeness and convergence criteria [13] and will depend upon the finite-element used for the spatial discretization.

Many previous works, Refs. [4], [11], and [12], utilized linear triangular shaped elements to discretize the spatial domain. This element was the first element considered. However, because the width of the cladding is very thin, elements in the cladding region would have extremely large aspect ratios (ratio of base to height) unless an extremely large number of elements in the axial direction were used. A large number of elements becomes numerically untractable. Previous experience with triangular elements had shown that large aspect ratios yield inaccurate results. Zlamal, Ref. [14], showed the error, e , when using triangular elements, is proportional to the square of the longest side, h , and inversely proportional to the sine of the smallest angle, γ

$$e \propto h^2/\sin\gamma$$

A triangular element with a large aspect ratio necessarily must have a small related angle which adversely affects the error in the FEM. Hopefully to alleviate the problem, an isoparametric, quadratic, rectangular element was selected. The aspect ratio would still be large, but experience, Ref. [15], with the use of rectangular elements indicated that a large aspect ratio is not always a detrimental factor.

The shape functions for this element are well documented, Ref. [13]. Utilizing a "local" coordinate system (See Figure 4), the shape functions may be written as

$$\begin{aligned} \text{Corner nodes} \quad i &= 1, 3, 5, 7 \\ N_i &= \frac{1}{4}(1+\xi_0)(1+\eta_0)(\xi_0+\eta_0-1) \end{aligned} \quad (38)$$

$$\text{Midside nodes} \quad N_i = \frac{1}{2}(1-\xi^2)(1+\eta_0) \quad i=2, 6 \quad (38a)$$

$$N_i = \frac{1}{2}(1+\xi_0)(1-\eta^2) \quad i=4, 8 \quad (38b)$$

where

$$\begin{aligned} \xi_0 &= \xi \xi_i \\ \eta_0 &= \eta \eta_i \end{aligned}$$

These normalized shape functions are shown in Figure 5.

The local and global coordinates are related by the following

$$r = \langle N_i \rangle^e \{r_i\}^e \quad (39)$$

and

$$z = \langle N_i \rangle^e \{z_i\}^e \quad (39a)$$

C. COORDINATE TRANSFORMATIONS

When using a local coordinate system, some simple transformations facilitate the integrations required by equation (37). In cylindrical coordinates with azimuthal symmetry

$$dV = 2\pi r dr dz \quad (40)$$

The derivative terms may be transformed by the following

$$\begin{Bmatrix} \frac{\partial N_i}{\partial r} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix} \quad (41)$$

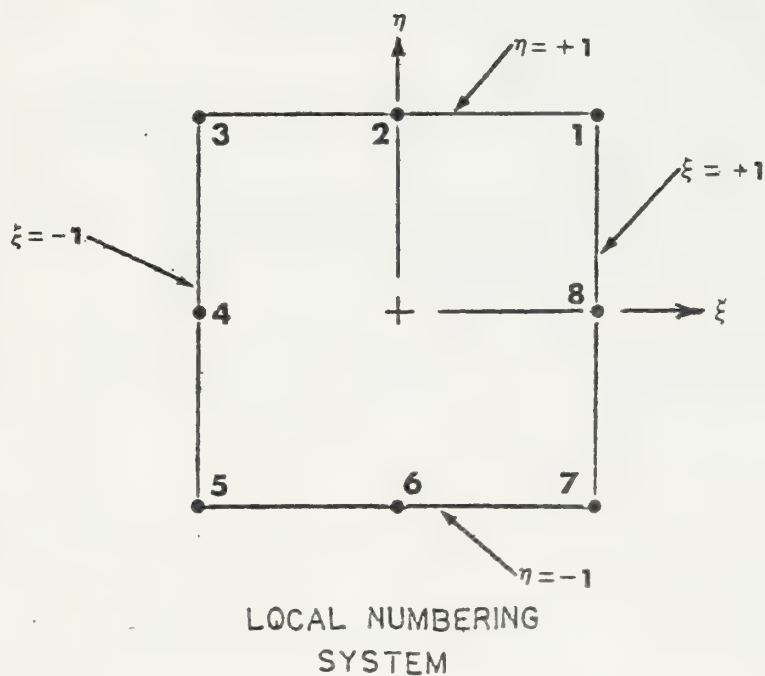
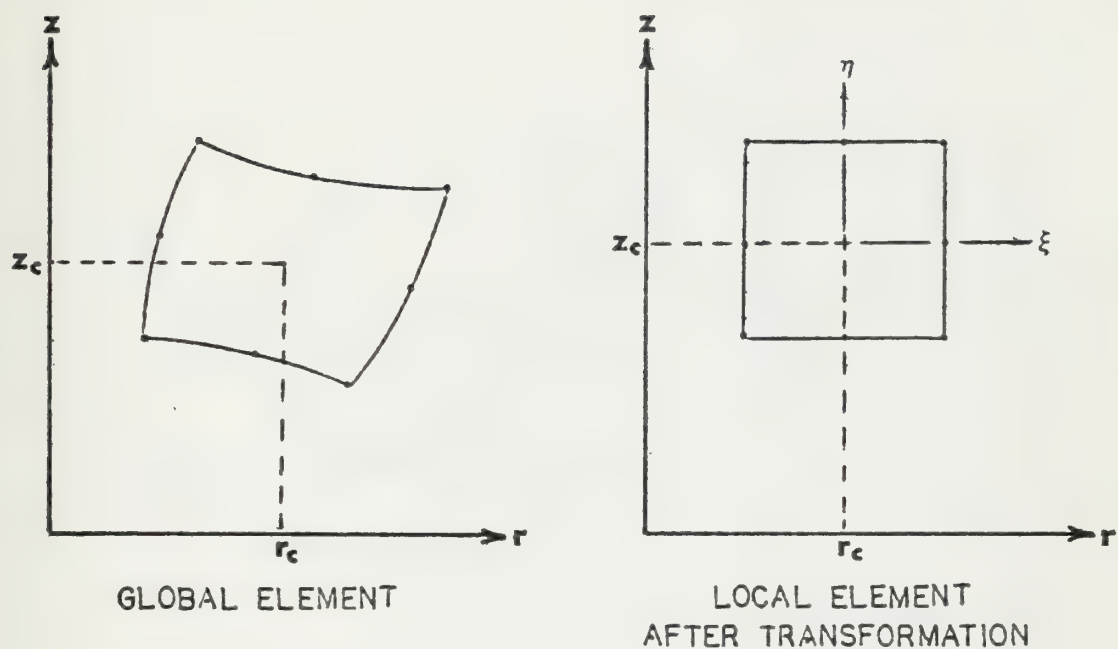
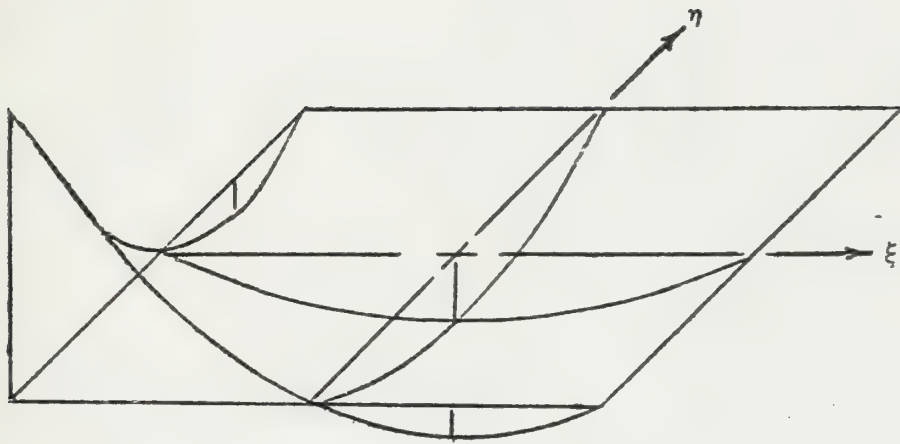
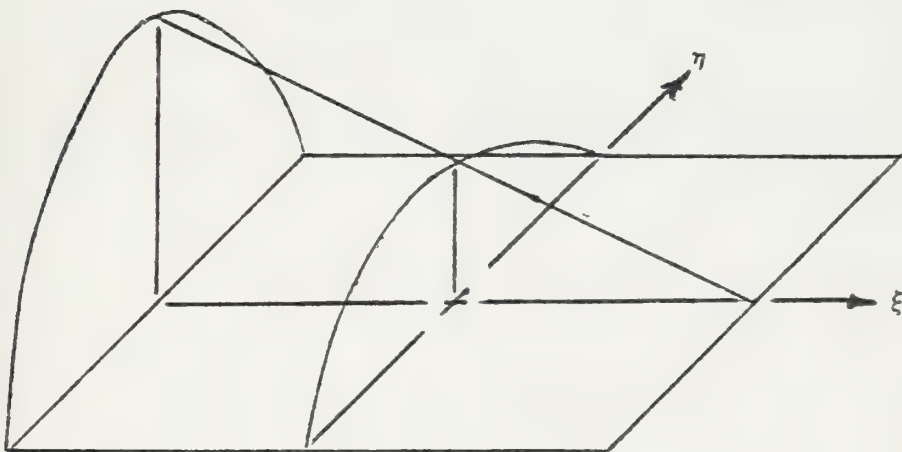


Figure 4. Element Transformation



CORNER NODE



MIDSIDE NODE

Figure 5. Normalized Shape Functions

where $[J]^{-1}$ is the inverse of the 2x2 Jacobian matrix defined in Appendix A. As shown in Appendix A, this inverse can be easily shown to be (for this problem)

$$[J]^{-1} = \begin{bmatrix} J_{11}^* & J_{12}^* \\ J_{21}^* & J_{22}^* \end{bmatrix} \quad (42)$$

where

$$\begin{aligned} J_{11}^* &= 2/\Delta r, & J_{12}^* &= 0 \\ J_{21}^* &= 0, & J_{22}^* &= 2/\Delta Z \\ \Delta r &- \text{radial length of the element} \\ \Delta Z &- \text{axial length of the element} \end{aligned}$$

Elements of area transform as

$$drdz = \det[J] \, d\xi d\eta \quad (43)$$

For this particular problem, $\det[J]$ may be shown to be (Appendix A)

$$\det[J] = \frac{A^e}{4} \quad (44)$$

where A^e - area of the element

Elements of axial length become

$$dz = \frac{L^e}{2} \quad (45)$$

where L^e - axial length of the element

Utilization of these transformations makes the integrations required in equation (37) amenable to integration by numerical Gaussian quadrature.

V. APPLICATION OF FEM TO GOVERNING FIELD EQUATIONS

In this chapter, equation (37) is applied to the governing field equations previously derived. The element matrices for each of the operators are developed so that the discretization of the spatial domain may be accomplished. The integrations required by the application of equation (37) are performed numerically using Gaussian quadrature.

A. GAUSSIAN QUADRATURE

Prior to the application of equation (37), it is appropriate to discuss briefly the procedure used for the numerical integration. The product Gaussian quadrature formula is [16]

$$I_A = \int_{-1}^1 \int_{-1}^1 g(\xi, \eta) d\eta d\xi = \sum_{i=1}^m \sum_{j=1}^m W_i W_j g(\xi_i, \eta_j) \quad (46)$$

where

I_A	- area integration
$g(\xi, \eta)$	- any function of ξ and η
$W_{i,j}$	- weight associated with location i or j
m	- number of Gauss sampling points in one-dimension

The values of the weights associated with each Gauss point are given in Ref. [16]. Equation (42) may be simplified somewhat by combining the summations and weights

$$I_A = \sum_{k=1}^{m^2} W_k g(\xi_k, \eta_k) \quad (47)$$

where

$$k = i \times j$$

$$W_k = W_i \times W_j$$

For line integrations, the Gaussian quadrature formula involves only one summation

$$I_L = \int_{-1}^1 f(\eta) d\eta = \sum_{i=1}^m W_i f(\eta_i) \tag{48}$$

B. NEUTRONIC FIELD EQUATIONS

The discretization of the spatial domain by the finite element method is accomplished by applying equation (37) to the governing field equations, using

$$\phi_k = \langle N_j \rangle^e \{ \psi_{kj} \}^e \qquad j=1,2,\dots,8 \tag{49}$$

$k=F,c,co$

1. Fuel Region

The governing equation for the fuel region, equation (20) after applying equation (37) becomes

$$\begin{aligned} &2\pi \int_r \int_z \frac{N_i}{v} \frac{\partial \phi_F}{\partial t} r dr dz - 2\pi \int_r \int_z N_i \left\{ \frac{1}{r} \frac{\partial}{\partial r} (r D_F \frac{\partial \phi_F}{\partial r}) \right. \\ &\quad \left. + \frac{\partial}{\partial z} (D_F \frac{\partial \phi_F}{\partial z}) + \Sigma_{aF} \phi_F (1-\beta) [k_{\infty}^o + \rho - b \ln(T_F/T_F^o)] \right. \\ &\quad \left. + \bar{\lambda} \bar{\beta} \Sigma_{aF} \int_0^t e^{-\lambda_i(t-t')} (k_{\infty}^o + \rho) \phi_F dt' + \bar{\lambda} C^o e^{-\lambda t} \right\} r dr dz = 0 \end{aligned} \tag{50}$$

The second order terms in equation (50) may be reduced to a first order term by application of Green's Theorem or equivalently integration-by-parts (See Appendix B). Dividing through by 2π and reducing the second order terms yields

$$\begin{aligned} & \int_z r N_i D_F \frac{\partial \phi_F}{\partial r} dz + \int_r r N_i D_F \frac{\partial \phi_F}{\partial z} dr + \int_r \int_z \frac{N_i}{v} \frac{\partial \phi_F}{\partial t} r dr dz \\ & + \int_r \int_z \left\{ D_F \left[\frac{\partial N_i}{\partial r} \frac{\partial \phi_F}{\partial r} + \frac{\partial N_i}{\partial z} \frac{\partial \phi_F}{\partial z} \right] - N_i \Sigma_{aF} \phi_F (1-\beta) [k_{\infty}^{\circ} + \rho - b \ln(T_F/F_{F0})] \right. \\ & \left. - N_i \bar{\lambda} \bar{\beta} \Sigma_{aF} \int_0^t e^{-\lambda(t-t')} (k_{\infty}^{\circ} + \rho) \phi_F dt' - N_i \bar{\lambda} C^{\circ} e^{-\bar{\lambda} t} \right\} r dr dz = 0 \quad (51) \end{aligned}$$

From continuity and boundary conditions the line integrals are zero. Now using the approximate functions of equation (49) and noting that $\{\psi_F\}^e$ is not a function of space, equation (51) may be written as

$$\begin{aligned} & \frac{1}{v} \iint_{r,z} \{N_i\}^e \langle N_j \rangle^e r dr dz \{\psi_F\}^e + \iint_{r,z} D_F [\{N_{i,r}\}^e \langle N_{j,r} \rangle^e + \{N_{i,z}\}^e \langle N_{j,z} \rangle^e] r dr dz \{\psi_F\}^e \\ & - \iint_{r,z} \Sigma_{aF} (1-\beta) [k_{\infty}^{\circ} + \rho - b \ln(T_F/T_F^{\circ})] \{N_i\}^e \langle N_j \rangle^e r dr dz \{\psi_F\}^e - \bar{\lambda} \bar{\beta} \Sigma_{aF} \int_0^t e^{-\bar{\lambda}(t-t')} (k_{\infty}^{\circ} + \rho) dt' \\ & \iint_{r,z} \{N_i\}^e \langle N_j \rangle^e r dr dz \{\psi_F\}^e - \bar{\lambda} C^{\circ} e^{-\bar{\lambda} t} \iint_{r,z} \{N_i\}^e r dr dz = 0 \quad (52) \end{aligned}$$

The integrations may not be easily carried out with a shift to local coordinates and with use of the previously derived transformations. Rearranging equation (52) and assuming the properties are constant for each time step gives

$$\begin{aligned}
 D_F \int_{-1}^1 \int_{-1}^1 & [\{N_{i,\xi} J_{11}^*\} \langle J_{11}^* N_{j,\xi} \rangle + \{N_{i,\eta} J_{22}^*\} \langle J_{22}^* N_{j,\eta} \rangle] \text{rdet}[J] d\xi d\eta \{\psi_F\}^e \\
 - \Sigma_{aF} (1-\beta) (k_\infty^\circ + \rho) & \int_{-1}^1 \int_{-1}^1 \{N_i\} \langle N_j \rangle \text{rdet}[J] d\xi d\eta \{\psi_F\}^e \\
 - \Sigma_{aF} (1-\beta) b & \int_{-1}^1 \int_{-1}^1 \ln \frac{T_F}{T_F^\circ} \{N_i\} \langle N_j \rangle \text{rdet}[J] d\xi d\eta \{\psi_F\}^e \\
 - \bar{\lambda} \bar{\beta} \Sigma_{aF} f(t) & \int_{-1}^1 \int_{-1}^1 \{N_i\} \langle N_j \rangle \text{rdet}[J] d\xi d\eta \{\psi_F\}^e - \bar{\lambda} C^\circ e^{-\bar{\lambda} t} \int_{-1}^1 \int_{-1}^1 \{N_i\} \text{rdet}[J] d\xi d\eta \\
 + \frac{1}{V} & \int_{-1}^1 \int_{-1}^1 \{N_i\} \langle N_j \rangle \text{rdet}[J] d\xi d\eta \{\dot{\psi}_F\}^e = 0
 \end{aligned} \tag{53}$$

Since the element chosen has eight degrees of freedom (nodal points), the discretized matrices which result from the integration of equation (53) will be 8x8 matrices and the forcing function will be an 8x1 vector at the element level. Defining the matrices as

$$\begin{aligned}
 & \int_{-1}^1 \int_{-1}^1 [\{N_{i,\xi} J_{11}^*\} \langle J_{11}^* N_{j,\xi} \rangle + \{N_{i,\eta} J_{22}^*\} \langle J_{22}^* N_{j,\eta} \rangle] \text{rdet}[J] d\xi d\eta \\
 = [H]_{2 \times 2} & = \frac{A^e}{4} \sum_{k=1}^m [(N_{i,\xi})_k (N_{j,\xi})_k J_{11}^{*2} \\
 & + (N_{i,\eta})_k (N_{j,\eta})_k J_{22}^{*2}] r_k W_k
 \end{aligned} \tag{54}$$

where $W_k = W_i W_j$ and $r_k = \sum_{i=1}^8 r_i (N_i)_k$

$$\int_{-1}^1 \int_{-1}^1 \{N_i\} \langle N_j \rangle r \det[J] d\xi d\eta = [H3_{ij}]_{8 \times 8} = \frac{A e^{m^2}}{4} \sum_{k=1}^8 (N_i)_k (N_j)_k r_k W_k \quad (55)$$

$$\int_{-1}^1 \int_{-1}^1 \{N_i\} r \det[J] d\xi d\eta = \{F_i\}_{8 \times 1} = \frac{A e^{m^2}}{4} \sum_{k=1}^8 (N_i)_k r_k W_k \quad (56)$$

$$\begin{aligned} \int_{-1}^1 \int_{-1}^1 \ln(T_F/T_D) \{N_i\} \langle N_j \rangle r \det[J] d\xi d\eta &= [H4_{ij}]_{8 \times 8} \\ &= \frac{A e^{m^2}}{4} \sum_{k=1}^8 \ln(T_F/T_{F^0})_k (N_i)_k (N_j)_k r_k W_k \end{aligned} \quad (57)$$

To carry out the summation of equation (57), the temperature T_F must be known; however, the temperature is exactly what is being sought. To alleviate this problem, a linearization is used.

In the solution technique, the temperature is predicted for the next time step. It is this temperature which is used for the determination of matrix $H4$.

Equation (53) simplifies to

$$\begin{aligned} \{D_F[H12_{ij}] - [\sum_{aF} (1-\beta)(k_\infty^0 + \rho) + \bar{\lambda} \bar{\beta} \sum_{aF} f(t)] [H3_{ij}] + \sum_{aF} (1-\beta)b [H4_{ij}]\} \{\psi_F\}^e \\ - \bar{\lambda} C^0 c^{-\bar{\lambda} t} \{F_i\}^e + \frac{1}{V} [H3_{ij}] \{\dot{\psi}_F\} = 0 \end{aligned} \quad (58)$$

The function $f(t)$ is evaluated by summing the values at each time step using the trapezoid rule for numerical integration.

$$f(t) = e^{-\bar{\lambda}t} \int_0^t e^{-\bar{\lambda}t'} [k_{\infty}^o + \rho] dt' \quad (59)$$

Defining

$$I_i[g(t)] = \frac{1}{2} h_i \{g(t_i) + g(t_{i-1})\} \quad (60)$$

$$g(t) = e^{\bar{\lambda}t} (k_{\infty}^o + \rho)$$

h_i - time step taken

$$g_0 = 0$$

The function may be expressed as

$$f(t) = e^{-\lambda t} \sum_{i=1}^S I_i [g(t)] \quad (61)$$

S - number of time steps

2. Clad Region

Following the same procedure as with the fuel equation, the discretized form of equation (11) becomes

$$\{D_c[H12_{ij}] + \Sigma_{ac}[H3_{ij}]\}\{\psi_c\}^e + \frac{1}{v}[H3_{ij}]\{\dot{\psi}_c\}^e = 0 \quad (62)$$

3. Coolant Region

The governing equation for the coolant region, equation (12), may be discretized into the form

$$\{D_{co}[H12_{ij}] + \Sigma_{aco}[H3_{ij}]\}\{\psi_{co}\}^e + \frac{1}{v}[H3_{ij}]\{\dot{\psi}_{co}\}^e = 0 \quad (63)$$

Once the governing equations have been discretized at the element level, they are combined into a system of equations at the global level. On the global level the governing equation for neutron transport takes the form of

$$[H]_{n \times n} \{\psi\}_{n \times 1} + [P]_{n \times n} \{\dot{\psi}\}_{n \times 1} + \{F\}_{n \times 1} = 0 \quad (64)$$

where $[H]$, $[P]$, and $[F]$ represent the system matrices and n is the number of nodal points used in the discretization. There are, then, n simultaneous ordinary differential equations used to describe the neutron transport problem.

C. HEAT TRANSPORT FIELD EQUATIONS

The spatial domain for the heat transport problem is discretized in the same manner as the domain for the neutronics problem. The same element matrices previously defined are valid. Let

$$T_k = \langle N_j \rangle^e \{\tau_{kj}\}^e \quad j=1,2,\dots,8 \quad (65)$$

$k=F,c,co$

1. Fuel Region

The governing field equation for the fuel, equation (24), is discretized by applying equation (37). Using an integration by part to lower the order of the second order terms allows equation (24) to be written as

$$\int_r \left[N_i k_F \frac{\partial T_F}{\partial z} \right]_0^z r dr dz + \int_z \left[N_i k_F \frac{\partial T_F}{\partial r} \right]_0^a dz - \int_r \int_z \left\{ k_F \left[\frac{\partial N_i}{\partial r} \frac{\partial T_F}{\partial r} + \frac{\partial N_i}{\partial z} \frac{\partial T_F}{\partial z} \right] \right. \\ \left. - e \Sigma_{fF} N_i \phi_F + N_i \rho_F C_F \frac{\partial T_F}{\partial t} \right\} r dr dz = 0 \quad (66)$$

From continuity considerations the line integrals are zero except along the boundaries of a region. It is assumed that no heat is transferred from the cell in the axial direction (boundary condition 5); therefore, the first line integral of equation (66) is zero. In the neutron diffusion problem there was continuity of flux at the interfaces so that the line integrals were zero; however, the heat transfer at the interfaces is affected by the gap and film conductances. The fuel-clad interface condition, equation (29), may be rewritten as

$$-k_F \frac{\partial T_F}{\partial r} = H_{\text{gap}}(T_F - T_C) = -k_C \frac{\partial T_C}{\partial r} \quad (67)$$

Substituting equation (67) into (66), dividing by minus one and utilizing equation (65) yields

$$\begin{aligned} & \int_z [r H_{\text{gap}} \{N_i\}^e \langle N_j \rangle^e]_0^a dz \{\tau_F\}^e - \int_z [r H_{\text{gap}} \{N_i\}^e \langle N_j \rangle^e]_0^a dz \{\tau_C\}^{e*} \\ & + \int_r \int_z k_F [\{N_{i,r}\}^e \langle N_{j,r} \rangle^e + \{N_{i,z}\}^e \langle N_{j,z} \rangle^e] r dr dz \{\tau_F\}^e \\ & - \int_r \int_z e \Sigma_{fF} \{N_i\}^e \langle N_j \rangle^e r dr dz \{\psi_F\}^e \\ & + \int_r \int_z \rho_F C_{pF} \{N_i\}^e \langle N_j \rangle^e r dr dz \{\dot{\tau}_F\}^e = 0 \end{aligned} \quad (68)$$

e^* - element across the interface

Transforming to local coordinates and integrating by Gaussian quadrature yields the same element matrices as given for the neutron flux, equations (54) and (55), except for the line integrals between regions. It should be noted that the line integrals exist only on the interfaces, along which there is a discontinuity of temperature. For the fuel equation the interface corresponds to the local coordinate $\xi=1$. Define

$$\int_{-1}^1 \{N_i\} \langle N_j \rangle d\eta = [k1]_{ij} 8 \times 8 = \frac{L^e}{2} \sum_{k=1}^{m^2} (N_i)_k (N_j)_k w_k \quad (69)$$

where the N 's are evaluated at $\xi=1$. Many of the terms of $K1$ will be zero since only the nodes on the $\xi=1$ boundary will have shape functions which are non-zero. It is through $K1$ that the temperatures for each region are coupled together. Equation (68) may now be written as

$$r_a H_{gap} \{ [K1] \{ \tau_F \}^e - [K1] \{ \tau_C \}^{e*} \} + k_F [H12] \{ \tau_F \}^e - e \Sigma_{fF} [H3] \{ \psi_F \}^e + \rho_F C_{pF} [H3] \{ \dot{\tau}_F \} = 0 \quad (70)$$

In obtaining this equation, it was assumed that material properties for each nodal point were constant at each time step. Perhaps a better assumption would have been to assume an average value for the properties of each element. The difference should not be significant, and the assumed constant nodal properties were numerically more tractable.

2. Clad Region

Applying equation (37) to the governing field equation for the clad, equation (25) gives the discretized form of the equation. Assuming no heat transfer in the axial direction on the boundaries (Boundary condition 5), equation (25) becomes

$$\begin{aligned}
 - \int_z [N_i r k_c \frac{\partial T_c}{\partial r} \Big|_a^b] dz + \int_r \int_z k_c \left[\frac{\partial N_i}{\partial r} \frac{\partial T_c}{\partial r} + \frac{\partial N_i}{\partial z} \frac{\partial T_c}{\partial z} \right] r dr dz \\
 + \int_r \int_z \rho_c C_{pc} N_i \frac{\partial T_c}{\partial t} r dr dz = 0
 \end{aligned} \quad (71)$$

In the cladding, there are two interfaces along which the line integral of equation (71) is not zero, along the fuel-clad interface and along the clad-coolant interface. For the fuel-clad interface, equation (61) applies. For the clad-coolant interface, the interface condition, equation (30), may be rewritten as

$$k_c \frac{\partial T_c}{\partial r} = h_{surf} (T_c - T_{co}) = -k_{co} \frac{\partial T_{co}}{\partial r} \quad (72)$$

Along the fuel-clad interface, the local coordinate corresponds to $\xi = -1$. Define the new element matrix

$$\int_{-1}^1 \int_{-1}^1 \{N_i\}^e \langle N_j \rangle^e d\eta = [K2_{ij}]_{8 \times 8} = \frac{L}{2} \sum_{k=1}^{m^2} (N_i)_k (N_j)_k w_k$$

where the N 's are evaluated at $\xi = -1$. As with $K1$, $K2$ will have many zero values because the shape functions are evaluated at $\xi = -1$.

Along the clad-coolant interface, the local coordinate corresponds to $\xi=1$ and the K1 matrix is appropriate.

Substituting equations (67) and (72), equation (71) becomes

$$\begin{aligned} & \int_z [r h_{\text{surf}} N_i (T_c - T_{co})]_b dz - \int_z [r h_{\text{gap}} N_i (T_F - T_c)]_a dz \\ & + \int_r \int_z k_c \left[\frac{\partial N_i}{\partial r} \frac{\partial T_c}{\partial r} + \frac{\partial N_i}{\partial z} \frac{\partial T_c}{\partial z} \right] r dr dz \\ & + \int_r \int_z \rho_c C_{pc} N_i \frac{\partial T_c}{\partial t} r dr dz = 0 \end{aligned} \quad (73)$$

The governing equation for the clad region may now be written as

$$\begin{aligned} & r_a h_{\text{gap}} \{ [K2] \{ \tau_c \}^e - [K2] \{ \tau_F \}^{e*} \} + r_b h_{\text{surf}} \{ [K1] \{ \tau_c \}^e - [K1] \{ \tau_{co} \}^{e*} \} \\ & + k_c [H12] \{ \tau_F \}^e + \rho_c C_{pc} [H3] \{ \dot{\tau}_c \} = 0 \end{aligned} \quad (74)$$

The line integrals of equation (73) affect only nodes which are on one of the boundaries; therefore, the nodal inputs into K1 and K2 are zero unless the node is on one of the boundaries.

3. Coolant Region

The field equation governing the coolant may be discretized in the same manner as above. After applying the Galerkin method and performing an integration by parts

on the second order terms, equation (25)

becomes

$$\begin{aligned}
 & -\int_z [r N_i k_{co} \frac{\partial T}{\partial r}]_b^c dz + \int_r \int_z k_{co} \left[\frac{\partial N_i}{\partial r} \frac{\partial T_{co}}{\partial r} + \frac{\partial N_i}{\partial z} \frac{\partial T_{co}}{\partial z} \right] r dr dz \\
 & + \int_r \int_z V_{co} \rho_{co} C_{pco} N_i \frac{\partial T_{co}}{\partial z} r dr dz \\
 & + \int_r \int_z \rho_{co} C_{pco} N_i \frac{\partial T_{co}}{\partial t} r dr dz = 0 \quad (75)
 \end{aligned}$$

The line integral, when evaluated at c, is zero (boundary condition 4). When evaluated at b, or correspondingly at $\xi = -1$, equation (72) is valid and K2 matrix is appropriate. All the terms of equation (75) have been defined except the flow term. Define

$$\begin{aligned}
 & \int_{-1}^1 \int_{-1}^1 \{N_i\}^e \langle N_{i,\eta} \rangle^e r \det[J] d\xi d\eta = [H5_{ij}]_{8 \times 8} \\
 & = \frac{A^e m^2}{4} \sum_{k=1}^2 (N_i)_k (N_{i,\eta})_k r_k w_k \quad (76)
 \end{aligned}$$

Transforming to local coordinates and integrating reduces equation (75) to

$$\begin{aligned}
 & r_b h_{surf} \{ [K2] \{ \tau_{co} \}^e - [K2] \{ \tau_c \}^{e*} \} + k_{co} [H12] \{ \tau_{co} \}^e \\
 & + V_{co} \rho_{co} C_{pco} [H5] \{ \tau_{co} \}^e + \rho_{co} C_{pco} [H3] \{ \dot{\tau}_{co} \}^e = 0 \quad (77)
 \end{aligned}$$

Now that the governing equations have been defined for each region on the element level, equations (70), (74), and (77), they may be assembled into a system equation on the global level. The equation will be in the general form of

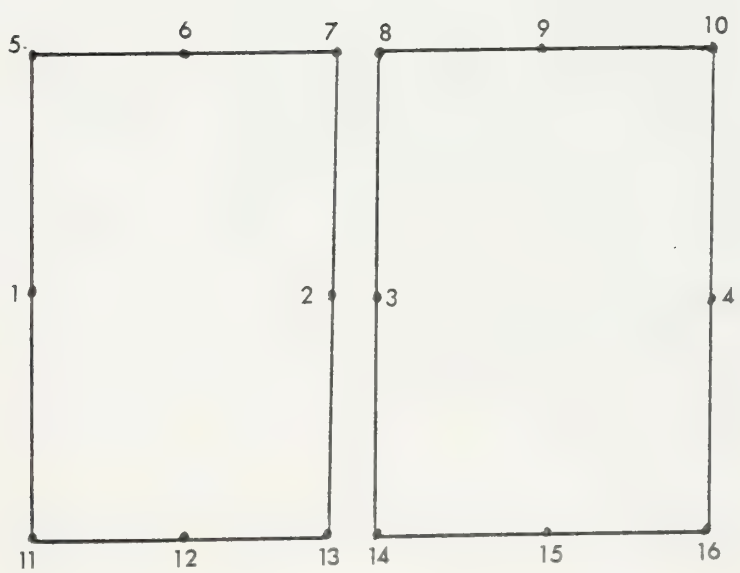
$$[K]_{n \times n} \{\tau\}_{n \times 1} + [M]_{n \times n} \{\psi\}_{n \times 1} + [G]_{n \times n} \{\dot{\tau}\}_{n \times 1} = 0 \quad (78)$$

D. DISCRETIZATION OF THE SPATIAL DOMAIN

Prior to the numerical solution of the governing equations, equations (64) and (78), the spatial domain must be divided into a number of elements. For this work the domain was subdivided as shown in figure 5.

Since there is a discontinuity of temperatures at the interfaces, as described by equations (29) and (30), a novel application of the FEM method was necessary. The common practice for handling these "flux" type boundary conditions is to define a constant reference temperature, T_∞ , as when working with a convection heat transfer problem [17], or to define a known function, as when working with a fracture mechanics problem [18]. In either case the reference condition was known. The novel application here lies in the use of a different field equation to describe the reference temperature, e.g., the clad equation (74) is the reference condition for heat transfer from the fuel across the gap interface.

The discontinuity of temperature at the interface necessitated another novel application of the FEM. Since there is a temperature drop along each interface, a single node there is not adequate. In the discretization of the domain, two nodes were used for each interface point (for example, points 62 and 63 in figure 6). This allows the temperature drop due to the gap and film conductances to be taken into consideration. Since two nodes are used, the governing equations for each region are not directly coupled together. The coupling of the regions is accomplished by the "flux" boundary or interface conditions since it is assumed that any heat flux leaving a region enters the adjacent region. Consider a typical set of elements on an interface



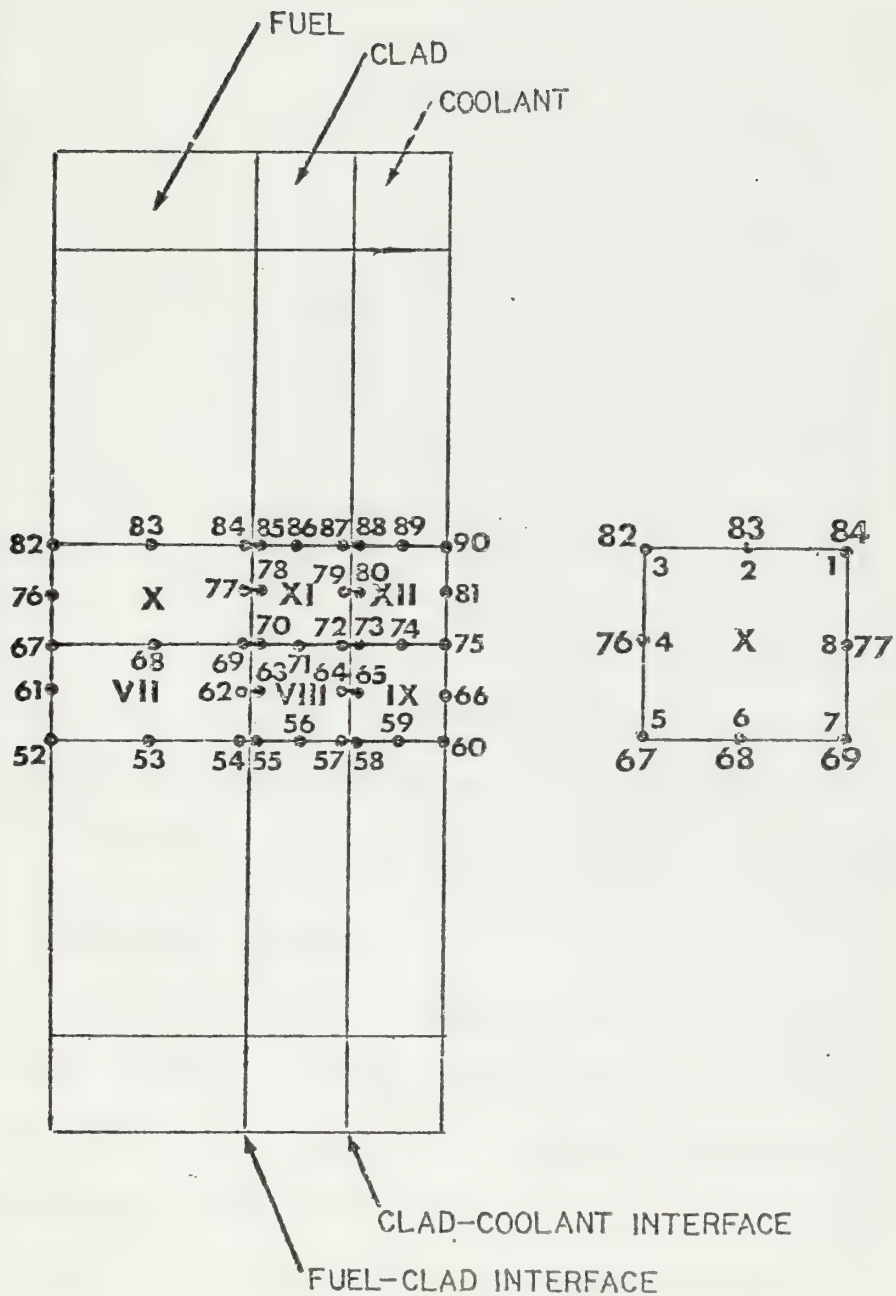


Figure 6. Finite Element Discretization

The coupling terms K1 and K2 may be combined into a system K matrix which shows the coupling. The K matrix for the simple set shown is

$$\begin{array}{c}
 \begin{array}{ccccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & \cdots & 13 & 14 & 15 & 16 \\
 \begin{array}{c}
 1 \\
 2 \\
 3 \\
 4 \\
 5 \\
 6 \\
 7 \\
 8 \\
 \vdots \\
 \vdots \\
 13 \\
 14 \\
 15 \\
 16
 \end{array}
 & \left[\begin{array}{cccccccccccc}
 & & & & & & & & & & & & & \\
 & \mathbf{a} & -\mathbf{a} & & & & \mathbf{c} & -\mathbf{c} & & & \mathbf{c} & -\mathbf{c} & & \\
 & -\mathbf{a} & \mathbf{a} & & & & -\mathbf{c} & \mathbf{c} & & & -\mathbf{c} & \mathbf{c} & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & \mathbf{c} & -\mathbf{c} & & & & \mathbf{b} & -\mathbf{b} & & & \mathbf{d} & -\mathbf{d} & & \\
 & -\mathbf{c} & \mathbf{c} & & & & -\mathbf{b} & \mathbf{b} & & & -\mathbf{d} & \mathbf{d} & & \\
 & \vdots & \vdots & & & & \vdots & \vdots & & & \vdots & \vdots & & \\
 & & & & & & & & & & & & & \\
 & \mathbf{c} & -\mathbf{c} & & & & \mathbf{d} & -\mathbf{d} & & & \mathbf{b} & -\mathbf{b} & & \\
 & -\mathbf{c} & \mathbf{c} & & & & -\mathbf{d} & \mathbf{d} & & & -\mathbf{b} & \mathbf{b} & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & &
 \end{array} \right]
 \end{array}
 \end{array}$$

$\mathbf{a} = 16/15$
 $\mathbf{b} = 4/15$
 $\mathbf{c} = 2/15$
 $\mathbf{d} = 1/15$

As can be seen, the nodes on the interface are coupled to the adjacent element interface nodes. For example, node 2 in element I is coupled to nodes 3, 8, and 14 in element II.

E. OPTIMUM COMPACTING SCHEME

The system matrices (H, P, etc.) are nxn matrices, where n is the number of nodal points used in the discretization of the domain. In terms of computer storage, these matrices may become excessively large if they are stored as nxn. There are several techniques available to reduce this storage requirement. The most common method is the banded storage scheme, whereby only the banded portion of the matrices are stored. With judicious numbering of the nodes,

The algorithm to assemble the element matrices into a compact storage vector is straightforward and represents a significant savings in computer storage [8]. The system matrices are stored as a vector rather than a two-dimensional array. For example, the value which would be stored in position (1,5) of the $n \times n$ array is stored in position 2 of the system vector.

VI. NUMERICAL SOLUTION

This section contains a brief description of possible solution techniques in addition to the solution technique chosen. Computer subroutines necessary to implement the technique are also described.

A. SELECTION OF METHOD

The numerical solution of the system of implicit ordinary differential equations, equations (64) and (78), may be accomplished by any of a number of different techniques such as Houbolt's method, Crank-Nicolson's method, Gear's method, or implicit Gear's method. It was not the objective of this analysis to determine which of the numerical solution schemes is the most efficient. Each method has its advantages and disadvantages. The Crank-Nicolson method is a single-step, implicit equation solver and, therefore, does not require storage of previous time solutions. When analyzing neutronic problems, the system of equations which arises is commonly very stiff (i.e., a rapid change in flux over a short period of time). The Crank-Nicolson method has, in a past work [8], demonstrated difficulty in tracking these stiff systems. Gear's method was specifically developed for stiff systems and can handle the problem very well. However, Gear's method is a multi-step, predictor-corrector method requiring storage of previous time solutions. In addition to this disadvantage, Gear's method requires the

transformation of the developed implicit O.D.E.'s into an explicit system of O.D.E.'s. After this transformation is done, the system matrices are no longer sparse or banded, thus eliminating the use of the optimum compacting scheme. In an effort to overcome these difficulties, Gear's method was modified, Ref. [9], to treat the implicit system of equations as well as to allow use of the optimum compacting scheme. A previous work, Ref. [8], has shown that the implicit Gear's method is particularly attractive in solving the type problem developed in this analysis. Therefore, the implicit Gear's method is used for the solution of the system of O.D.E.'s arising in this analysis.

No attempt will be made here to give the mathematics involved in developing the implicit Gear's method. Reference [9] may be consulted if details are desired. A listing of the computer program developed will be given in the Computer Program section. In order to utilize the implicit Gear's method, several user supplied subroutines must be developed: 1) DIFFUN, 2) JACMAT, and 3) NUITSL.

B. USER SUPPLIED SUBROUTINES TO IMPLEMENT THE IMPLICIT GEAR'S METHOD

1. DIFFUN

Subroutine DIFFUN evaluates equations (64) and (78) for a given time and for given values of ψ , $\dot{\psi}$, τ and $\dot{\tau}$. Since at each nodal point, i , there is a solution for the flux and for the temperature, the solution was set equal to DYI and DYII, respectively. In addition to having flux and

temperature at each nodal point, there are also three different regions in the domain which have different governing equations. An integer array, ITYPE, was developed to indicate for each nodal point whether it was: 0) a fuel node not in an interface element, 1) a fuel node in an interface element, 2) a cladding node or, 3) a coolant node. Using ITYPE, the computer program is directed to a different section depending upon the type of node being considered. After all the nodes have been considered, boundary conditions are established by changing DYI and DYII for the appropriate boundary nodes. Since in this analysis there is continuity of flux at the interfaces, special considerations must be given to these nodes. At the fuel-clad interface, the value of DYI for the clad node was set to the value of the flux at that node minus the value of the flux at the adjacent node (i.e., $DYI_i = \psi_i - \psi_{i-1}$). Similarly, at the clad-coolant interface, the value of DYI for the coolant node was set to the value of the flux at that node minus the value of the flux at the adjacent node. During the solution of the problem, DYI is driven toward zero, which in the limit forces ψ_i to equal ψ_{i-1} . This is the desired continuity result.

2. JACMAT

Subroutine JACMAT, evaluates the Jacobian matrix (for Gear's method) at the given time and for the current values of the dependent variables. The Jacobian for an equation of the type,

$$F(y, \dot{y}, t) = 0 \quad (79)$$

may be represented as [19],

$$J = \left[\frac{\partial F}{\partial y} - \frac{\alpha_0}{\beta_0 h} \frac{\partial F}{\partial \dot{y}} \right] \quad (80)$$

where α_0 and β_0 are coefficients from Gear's method and h is the time step. Using the notation of DIFFUN, let DYI and DYII represent equations (64) and (78), respectively. The Jacobian matrix may, then, be written as (J is called PW in JACMAT.)

$$PW = \left[\frac{\partial DYI}{\partial \psi} - \frac{\alpha_0}{\beta_0 h} \frac{\partial DYI}{\partial \dot{\psi}}, \frac{\partial DYII}{\partial \tau} - \frac{\alpha_0}{\beta_0 h} \frac{\partial DYII}{\partial \dot{\tau}} \right] \quad (81)$$

It is the form of equation (81) which is programmed in JACMAT. As in DIFFUN, the integer array ITYPE is used to indicate the appropriate section of the program to be utilized. The problem boundary conditions must also be accounted for in JACMAT. In DIFFUN, the value of DYI or DYII was set to zero for constant boundary conditions (i.e., zero). This cannot be done in JACMAT since a division by zero would occur. For a constant boundary condition at the i th node, the value of PW is set to one for the diagonal term and zero for all other terms of the i th equation.

3. NUITSL

Subroutine NUITSL solves the system of equations for the quasi-Newton iterates. In this analysis the system is solved using a successive over-relaxation (SOR) method. In this work, the optimum amount of over-relaxation was not determined. Since no effort was made to find the optimum,

it was felt a small over-relaxation would be best. The over-relaxation factor of 0.02 was used. For small values of this factor, the SOR method approaches the Gauss-Siedel iteration technique.

VII. PROCEDURE

In this section, the method utilized to obtain a solution is described. The input data necessary to run the developed computer program will be documented.

Prior to initiating a transient overpower excursion, the steady-state conditions for the fuel cell must be known. Since the system of equations which were developed are not specifically designed to obtain a steady-state solution, the initial steady-state conditions must be part of the input data. The initial temperature distribution was obtained from the steady-state conditions given in Ref. [7].

The axial temperature distribution for the fuel centerline, fuel surface, clad, and coolant have been determined for several different fuel life cycles [7]. For this analysis, the beginning of life cycle for channel 10 was used. Although this distribution is somewhat artificial, it should be adequate for this analysis. It is the trends of the results which are considered important. The distribution within the fuel radially is taken to vary as the square of the radial distance; then

$$T_F(r,z,0) = T_F(0,z,0)\left(1 - \frac{r^2}{a^2}\right) + T_F(a,z,0)\left(\frac{r^2}{a^2}\right)$$

Within the cladding and the coolant, the initial radial temperature distribution is assumed to be constant.

The initial flux distribution is assumed to be radially constant, a flat flux assumption. In the axial direction the flux is assumed to vary as the shape of the sine function. The maximum flux, the flux at the axial center, is an input parameter. For this analysis, the maximum initial flux was taken to be 10^{14} neutron/cm²sec.

To obtain a steady state flux distribution, the value of fission cross section for the fuel is varied. A trial-and-error method is used until a critical fission cross section, Σ_{fF}^{cr} , which gives a steady flux is obtained.

Once the steady-state conditions have been determined, the excess reactivity may be inserted. This starts the transient overpower excursion.

A. INPUT DATA

The first data card contains: the order of Gauss quadrature, the number of radial elements in the fuel, the number of axial elements, number of nodal points in the radial direction, and the height of the fuel rod. The next cards, one for each radial nodal point, contain the nodal radial distances. The next cards contain the fuel centerline, fuel surface, clad and coolant temperatures. There is one card for each axial node. The next card contains the maximum flux. The next four cards contain the physical parameters listed in Table I. The last input cards contain the time, end time, estimated initial time step, minimum time step, and maximum time step. A sample data deck is shown in figure 7.

TABLE I. Physical Parameters

Fuel Diffusion Coefficient (DCF)	0.93 [cm]
Doppler constant (B)	0.006
Energy release per fission (E)	7.652×10^{-12} [cal/fissions]
Fraction of delayed neutrons (BETA)	0.0064
Fraction of delayed neutrons for the ith group (BETAI)	0.0064
Decay constant for ith delayed neutron group (DCLAMI)	0.0784
Initial flux for delayed neutrons	1×10^{10} [neutron/cm ² sec]
Average number of neutrons released per fission (ANU)	2.44
Neutron velocity (VEL)	4.8×10^8 [cm/sec]
Fuel absorption cross section (SIGAF)	0.088 [cm ⁻¹]
Critical fuel fission cross section (SIGFF)	0.0586875 [cm ⁻¹]
Blanket absorption cross section (SIGAB)	0.0 [cm ⁻¹]
Blanket fission cross section (SIGFB)	0.0 [cm ⁻¹]
Step reactivity input (RHOA)	Variable
Ramp reactivity input (RHOB)	Variable
Fuel density (DENF)	10.9 [gm/cm ³]
Clad diffusion coefficient (DCC)	1.1 [cm]
Clad specific heat (CPC)	0.12 [cal/gm °C]
Clad density (DENC)	8.0 [gm/cm ³]
Clad thermal conductivity (TKC)	0.0526 [cal/cm sec °C]
Clad absorption cross section (SIGAC)	0.0015 [cm ⁻¹]
Coolant diffusion coefficient (DCCO)	1.55 [cm]
Coolant absorption cross section (SIGACO)	0.00004 [cm ⁻¹]
Coolant flow velocity (VCO)	396.0 [cm/sec]
Surface heat transfer coefficient(HSURF)	0.7 [cal/cm ² sec °C]

VIII. RESULTS

When using the finite element method, one of the first considerations must be given to the convergence of the method. To determine convergence, the results for a given point are compared for different finite element discretizations. As shown in figure 8, the results are comparable but no definite claim of convergence can be made. However, for this work, it was felt that these results were adequate. It was not the object of this analysis to arrive at the "final" result; it was the trends and methods that were of interest. Since the 66-element mesh appears to give a fair approximation of the results, the 66-element mesh was used as the discretized domain.

The next item of consideration was the determination of a neutronic steady-state condition. This proved to be a very time consuming task. The fission cross section for the fuel was varied by a trial-and-error method in an attempt to find the critical cross section which would give a steady state. As may be seen in figure 9, a change in cross section of less than one percent significantly affected the state of the problem. It was felt that the critical value was between the values of 0.05875 and 0.058625. Time did not permit investigation for critical value; therefore, it was assumed that the value for the critical fission cross section was half way between the values (i.e., $\Sigma_f^{cr} = 0.0586875$).

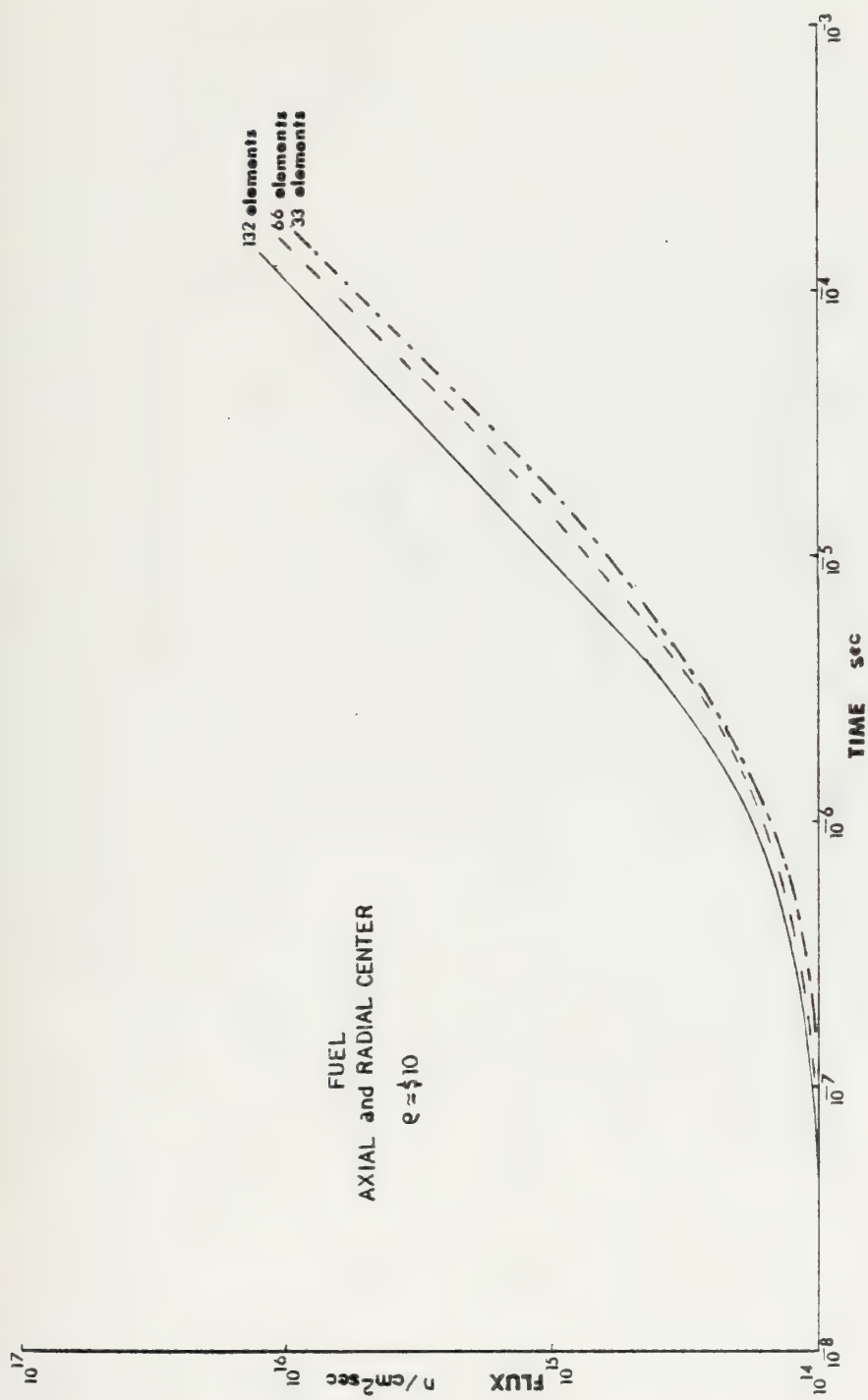


Figure 8. Convergence of Finite Element Method

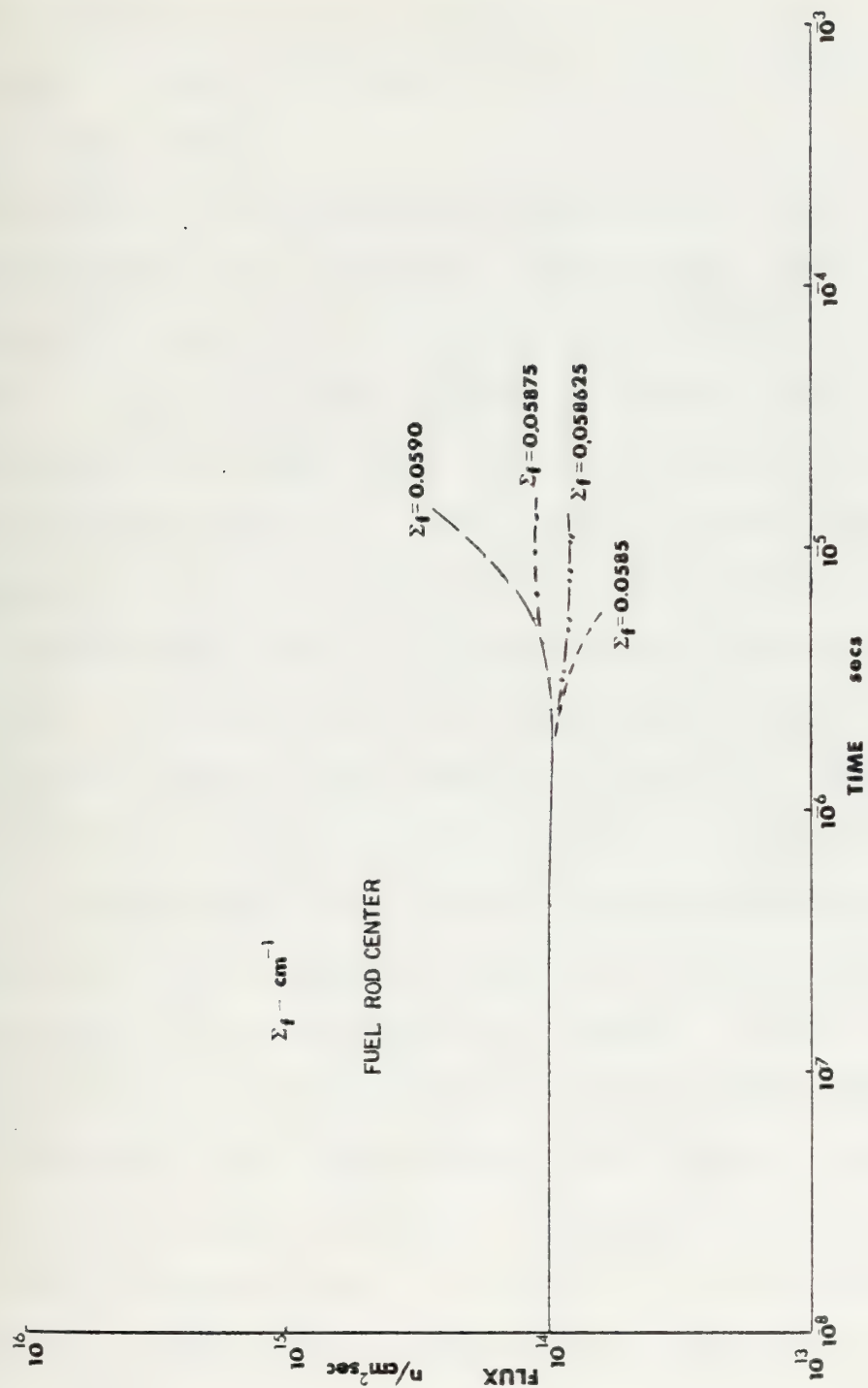


Figure 9. Determination of Critical Fission Cross Section

Even if this value is in error, which it most probably is, the net effect would only be a small decrease or increase in the proposed reactivity insertion. The reactivity insertion would, then, be an approximation of the actual reactivity of the problem.

The first test problem considered was a step increase in reactivity of approximately ten dollars. For the uranium dioxide fuel, one dollar of reactivity was taken to be 0.0064. Figure 10 shows the time history of the flux at the center of the fuel rod. Figure 11 gives the corresponding temperature profile. The temperatures were taken at the hottest point of each region at the axial center (i.e., the fuel centerline, clad inside surface, and coolant inside surface). As seen on the fuel temperature time history, the fuel rapidly reaches the fuel melting point. The model developed does not take into consideration melting of the fuel. This melting would tend to decrease the effect of the transient. The problem was allowed to continue despite this inconsistency in the mathematical model. A short time after fuel melting, the inside surface of the clad reaches its melting point. The temperature in the coolant experienced what is felt to be a numerical phenomenon. The coolant temperature decreased prior to the small rise at the end of the transient. Intuitively, this decrease does not seem to be realistic. A similar occurrence was observed while conducting sample tests on the developed computer program. Reference [20] reported the same phenomenon. It is felt that this

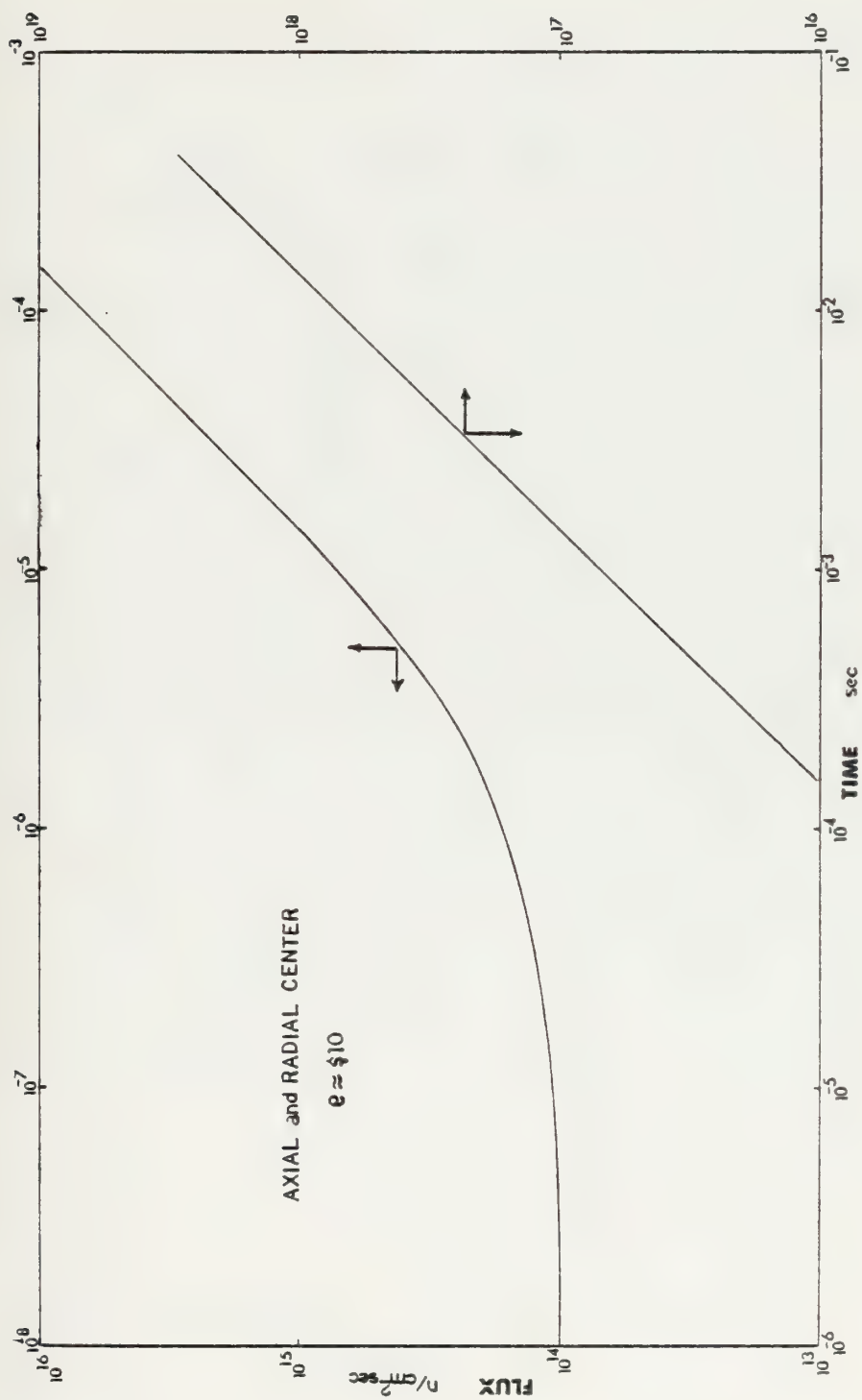


Figure 10. Flux Profile

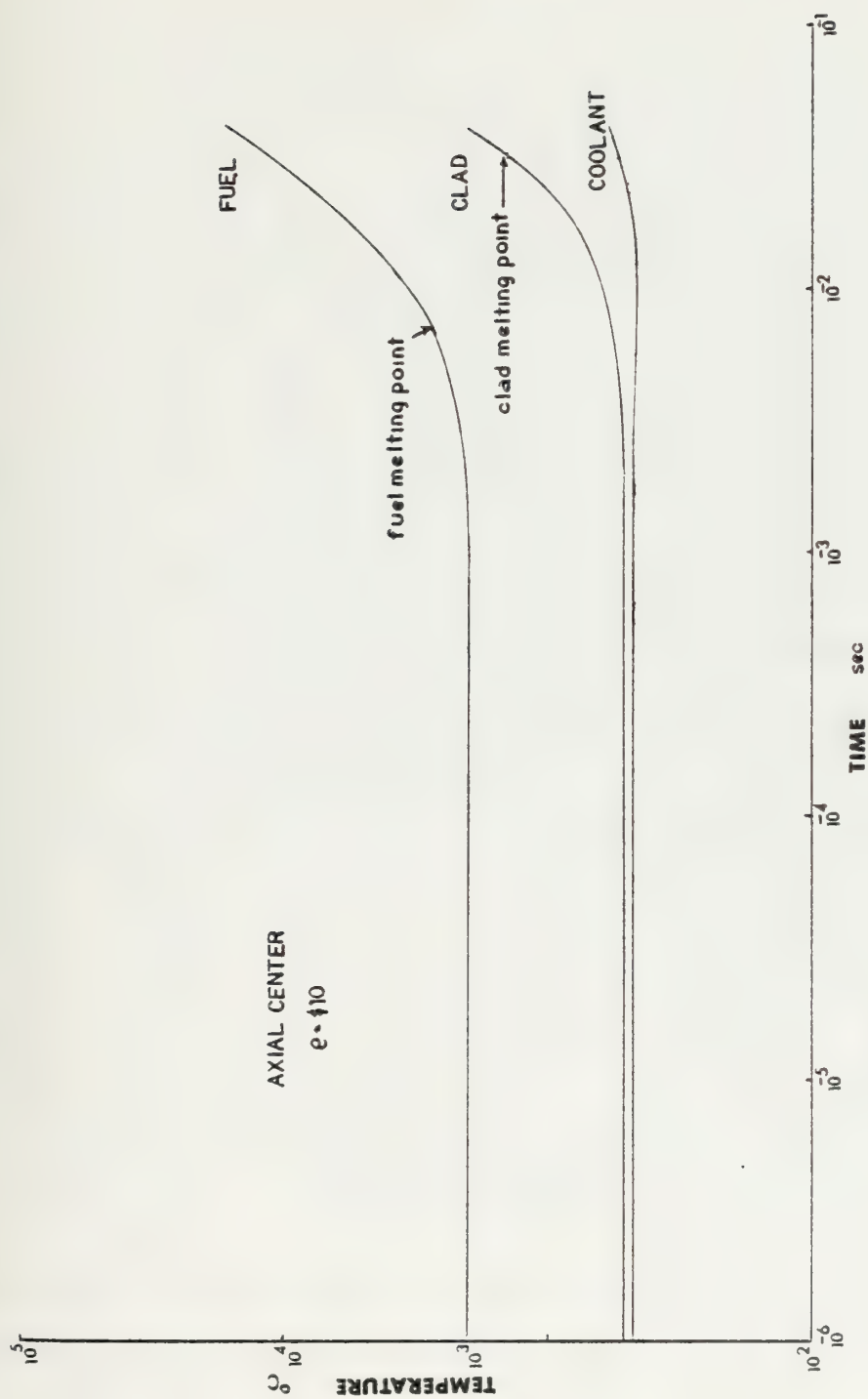


Figure 11. Temperature Profile

phenomenon is a quirk of the finite element method. The radial and axial distributions of the neutron flux are presented in figures 12 and 13 for time equal to 7.39×10^{-3} seconds. The distributions are, basically, as anticipated. The neutron flux peaks slightly before the axial center. It was expected to peak at the axial center. The radial and axial temperature distributions for the same time are presented in figures 14 and 15. As with the flux, the temperature profiles were, basically, as expected. The fuel temperature peaks slightly below the expected location, most likely in response to the peak in axial fluxes. The coolant unexpectedly drops near the outlet of the fuel rod. The finite element method characteristically has some problems on the boundaries of the domain; this may account for the drop in coolant temperature.

The temperature of the fuel and cladding do not appear to be as closely coupled as anticipated. As seen in figure 14, a significant increase in fuel temperature has resulted in a relatively small clad temperature increase. As noted in figure 11, there appears to be a time lag in temperature response for each region which may account for part of the apparent temperature disagreement. It is felt that the temperatures should be more closely related, which indicates a higher gap heat transfer coefficient should be utilized. Since values of the gap heat transfer coefficient were assumed, it is not unreasonable to believe the values used are too low.

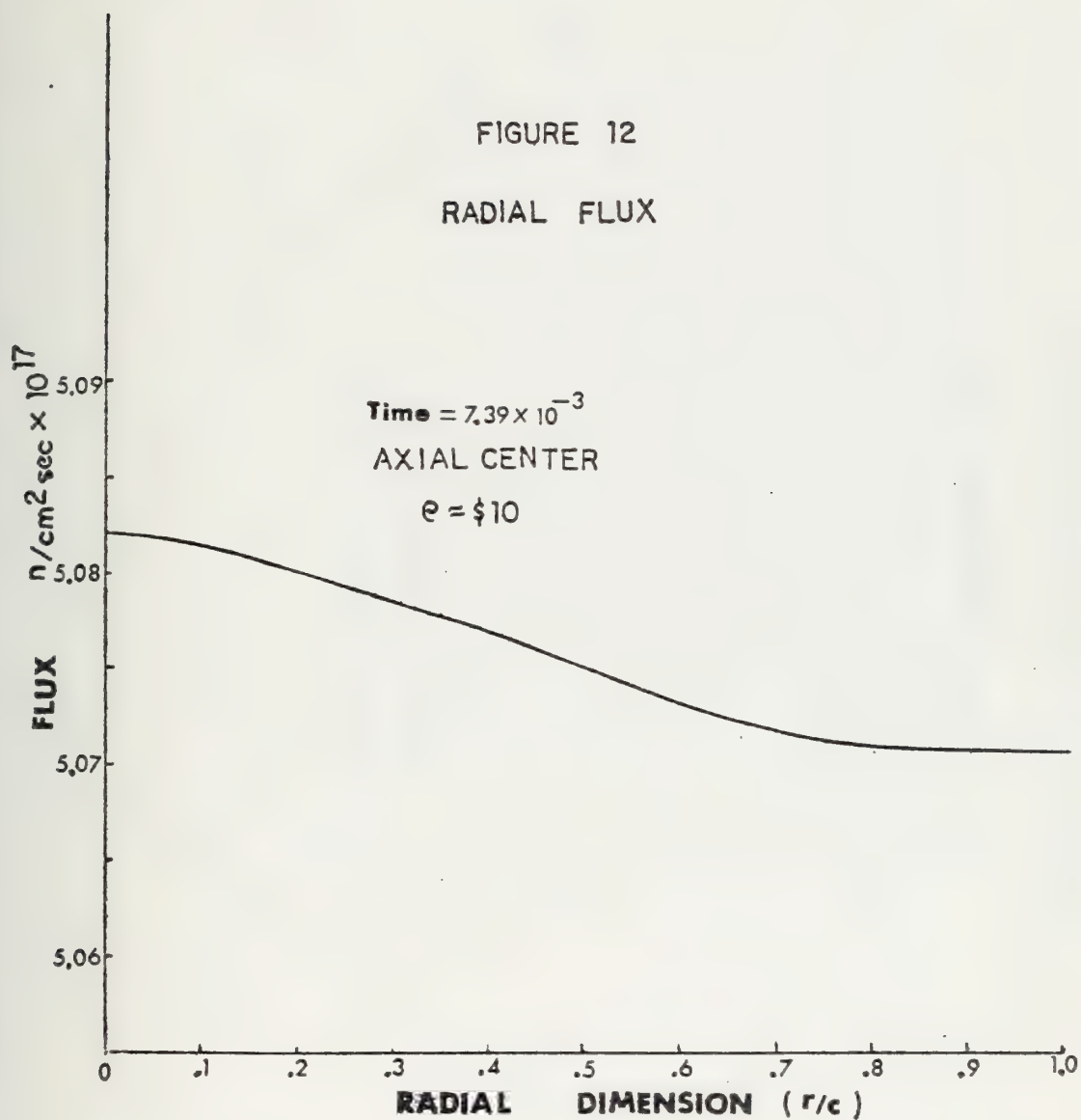


Figure 12. Radial Flux

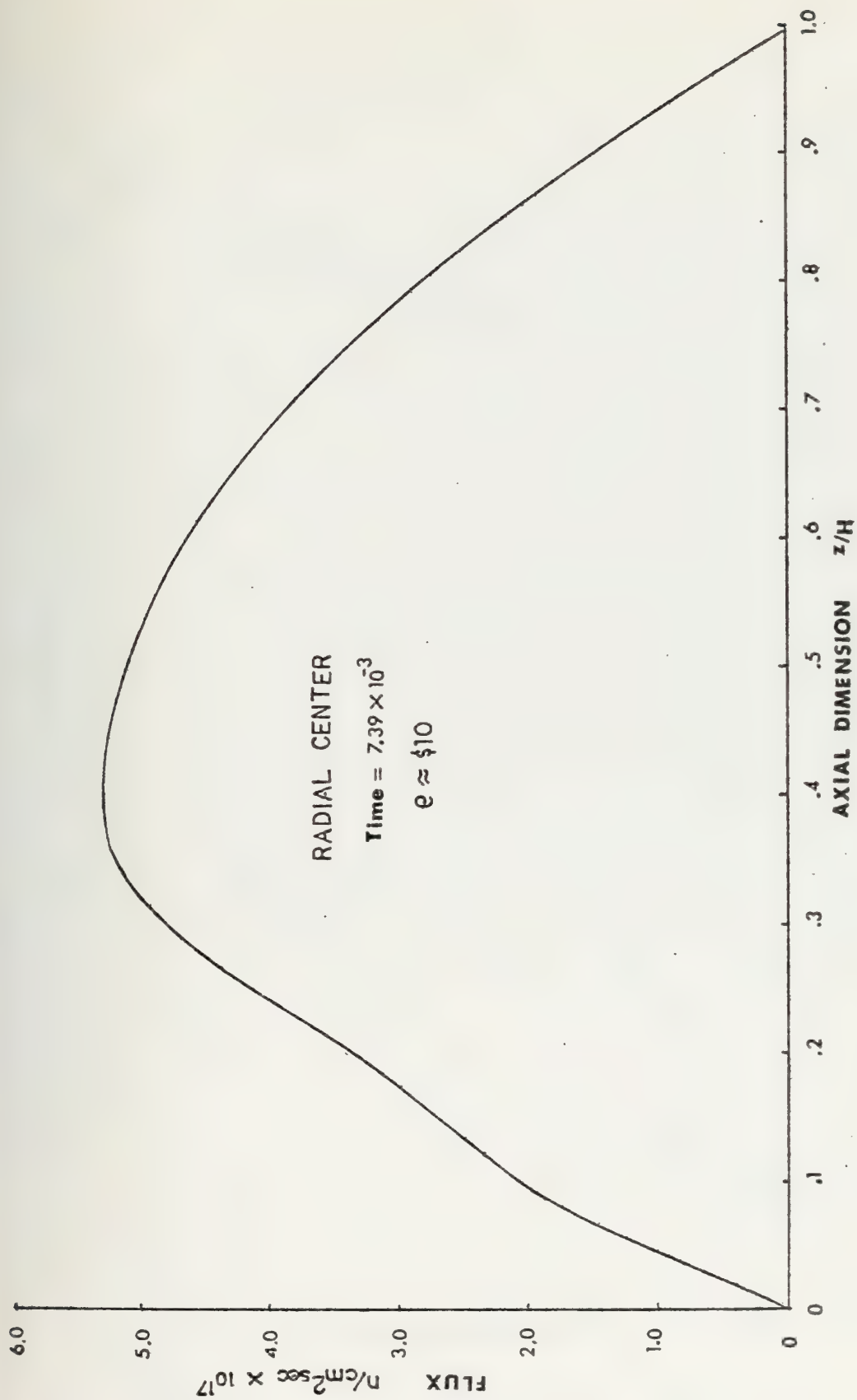


Figure 13. Axial Flux Profile

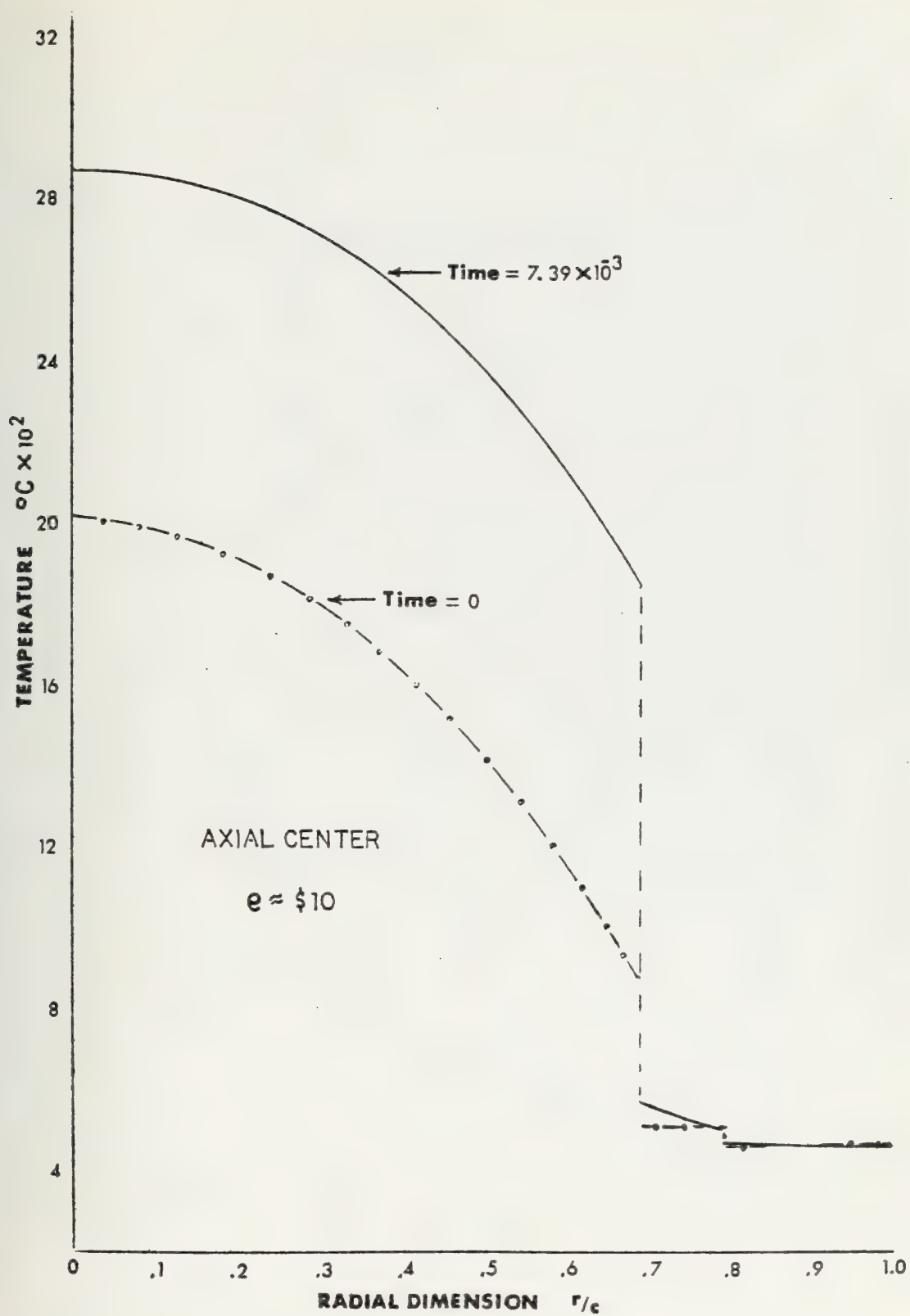


Figure 14. Radial Temperature Profile

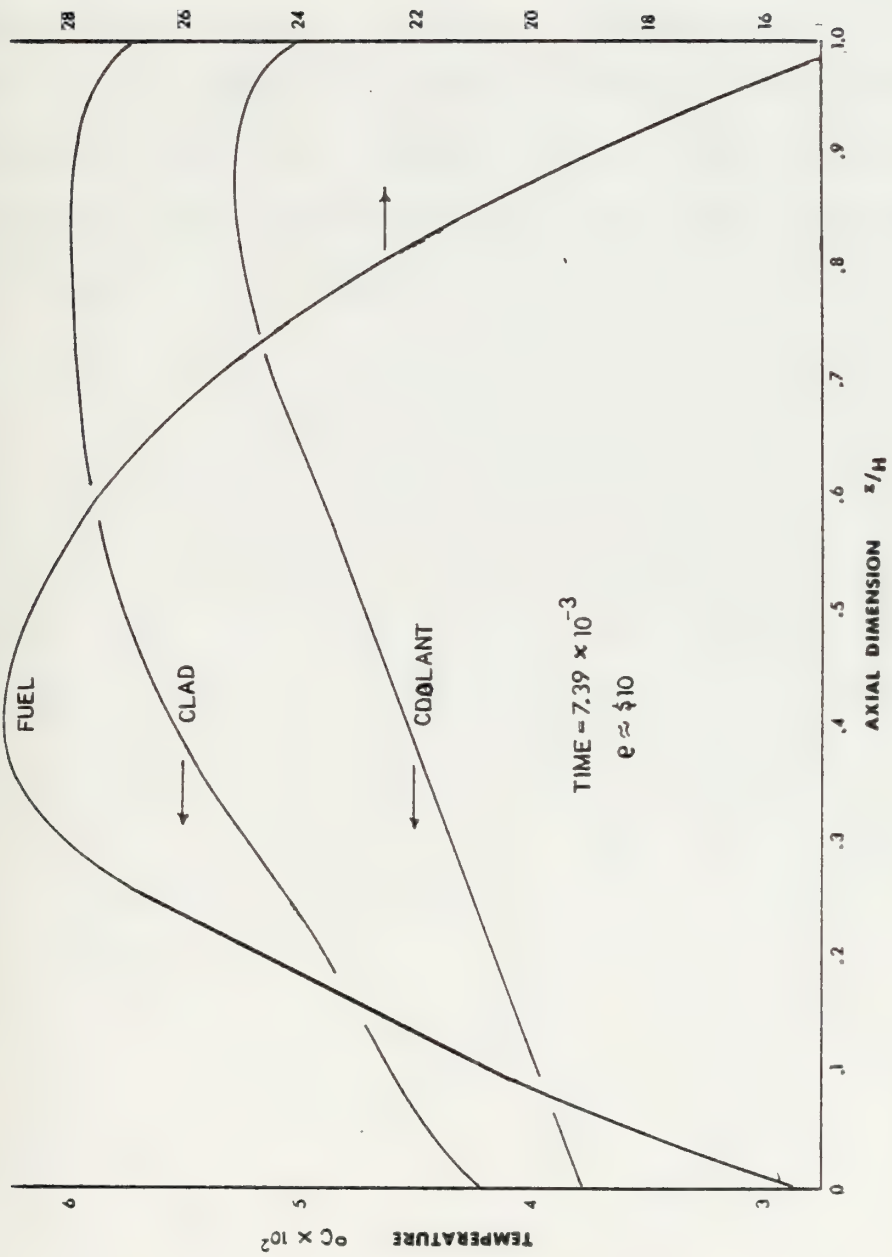


Figure 15. Axial Temperature Profile

Time did not permit investigation of other reactivity insertions. Other reactivity inputs may be investigated by students in the future.

This work does not represent a solution to the very complicated nuclear reactor problem. It does represent an application of a numerical technique which is relatively new to nuclear applications. Methods for implementing the finite element method have been discussed, and a computer code has been developed for the simplistic model considered.

IX. RECOMMENDATIONS

For the model developed, perhaps the most important item to pursue is the critical fission cross section. A better determination of this value is necessary so that the reactivity insertion is more accurately known. Different test cases for the prompt critical and prompt subcritical reactor could then be conducted.

In further developing the model, more consideration should be given the gap heat transfer coefficient. As noted in the results, the value used appears to be too small. Sample problems for different gap heat transfer coefficients would give a better indication of the values to use.

Melting of the fuel during the transient would probably be the next major improvement on the model. With relatively few changes, the model could be adapted to allow melting element by element. This, too, would be an approximation but, still, an improvement to the model. Perhaps at the same time, a simplified model to take into consideration the fuel restructuring could be implemented.

Another improvement would be to consider reactivity feedbacks in addition to the Doppler feedback. Sodium voiding and fuel rod expansion are two of the more important feedback effects to consider.

On the numerical side, probably the most important thing to do would be to run the computer program on the

"H-compiler", which optimizes the program. However, on several runs using the H-compiler, erroneous results were obtained. With sufficient time, this could be corrected to allow use of the H-compiler. The use of the H-compiler results in a savings in computer time. The present program runs on a "G-compiler" and takes excessive amounts of computer time (two to four hours per run).

In addition to this, the optimum over-relaxation factor in the implicit Gear's method could be determined by trial-and-error.

Implementation of these recommendations should enhance the analysis and lead to a more efficient computer code.

APPENDIX A
DEVELOPMENT OF TRANSFORMATIONS

The Jacobian matrix $[J]$ may be written (8) for two dimensions as

$$[J] = \begin{bmatrix} \sum_{i=1}^N N_{i,\xi} r_i & \sum_{i=1}^N N_{i,\xi} z_i \\ \sum_{i=1}^N N_{i,\eta} r_i & \sum_{i=1}^N N_{i,\eta} z_i \end{bmatrix} \quad (A1)$$

For a simple 2x2 matrix $[A]$ the inverse is

$$[A] = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$[A]^{-1} = \frac{1}{\det[A]} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Applying this fact to equation (A1) gives

$$[J]^{-1} = \frac{1}{\det[J]} \begin{bmatrix} \sum_{i=1}^N N_{i,\eta} z_i & - \sum_{i=1}^N N_{i,\xi} z_i \\ \sum_{i=1}^N N_{i,\eta} r_i & \sum_{i=1}^N N_{i,\xi} r_i \end{bmatrix} \quad (A2)$$

$$= \begin{bmatrix} J_{11}^* & J_{12}^* \\ J_{21}^* & J_{22}^* \end{bmatrix}$$

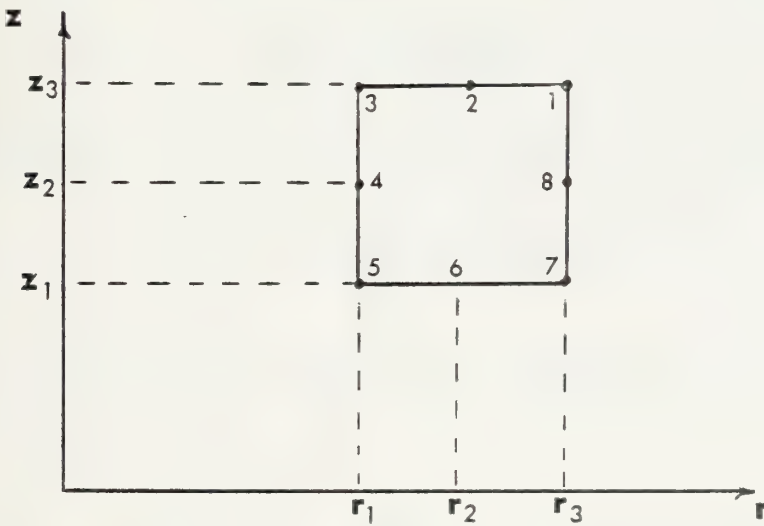
From matrix algebra

$$\det[J] = \sum_{i=1}^N N_{i,\xi} r_i \sum_{i=1}^N N_{i,\eta} z_i - \sum_{i=1}^N N_{i,\eta} r_i \sum_{i=1}^N N_{i,\xi} z_i \quad (A3)$$

The derivatives of the shape functions may be found from equations (38).

$$\begin{aligned} N_{1,\xi} &= \frac{1}{4}(1+\eta)(2\xi+\eta) & N_{1,\eta} &= \frac{1}{4}(1+\xi)(2\eta+\xi) \\ N_{2,\xi} &= -\xi(1+\eta) & N_{2,\eta} &= \frac{1}{2}(1-\xi^2) \\ N_{3,\xi} &= \frac{1}{4}(1+\eta)(2\xi-\eta) & N_{3,\eta} &= \frac{1}{4}(1-\xi)(2\eta-\xi) \\ N_{4,\xi} &= -\frac{1}{2}(1-\eta^2) & N_{4,\eta} &= -\eta(1-\xi) \\ N_{5,\xi} &= \frac{1}{4}(1-\eta)(2\xi+\eta) & N_{5,\eta} &= \frac{1}{4}(1-\xi)(2\eta+\xi) \\ N_{6,\xi} &= -\xi(1-\eta) & N_{6,\eta} &= -\frac{1}{2}(1-\xi^2) \\ N_{7,\xi} &= \frac{1}{4}(1-\eta)(2\xi-\eta) & N_{7,\eta} &= \frac{1}{4}(1+\xi)(2\eta-\xi) \\ N_{8,\xi} &= \frac{1}{2}(1-\eta^2) & N_{8,\eta} &= -\eta(1+\xi) \end{aligned} \quad (A4)$$

If one now considers an arbitrary element



with the midside nodes exactly at the midpoint (not a necessary criteria for the FEM) and substitutes into equation (A3),

the result is

$$\det[J] = \frac{(z_3 - z_1)(r_3 - r_1)}{4} = \frac{A^e}{4} \quad (A5)$$

Substituting into equation (A2) and using (A5) will yield

$$J_{11}^* = \frac{4}{A^e} \left[\frac{1}{2}(z_3 - z_1) \right] = 2/r_3 - r_1, \quad (A6)$$

$$J_{12}^* = 0, \quad (A7)$$

$$J_{21}^* = 0, \quad (A8)$$

and
$$J_{22}^* = \frac{4}{A^e} \left[\frac{1}{2}(r_3 - r_1) \right] = 2/z_3 - z_1. \quad (A9)$$

The inverse of the Jacobian matrix now becomes

$$[J]^{-1} = \begin{bmatrix} 2/r_3 - r_1 & 0 \\ 0 & 2/z_3 - z_1 \end{bmatrix} \quad (A10)$$

For integration along a line, the transformation used is

$$dz = \det[J'] d\eta \quad (A11)$$

In this case

$$\det[J'] = \sum_{i=1}^N N_{i,\eta} z_i \quad (A12)$$

Again considering the arbitrary element and substituting (A4) into (A12) will result in

$$\det[J'] = \frac{z_3 - z_1}{2} = \frac{L^e}{2} \quad (A13)$$

APPENDIX B

REDUCTION OF SECOND ORDER TERM

The second order term of the governing field equations may be reduced to first order by integration by parts.

Consider, for example,

$$\iint_{r,z} N_i \left[\frac{1}{r} \frac{\partial}{\partial r} (rD \frac{\partial \psi}{\partial r}) + \frac{\partial}{\partial z} (D \frac{\partial \psi}{\partial z}) \right] r dr dz \quad (B1)$$

which may be expanded as

$$\iint_{r,z} [N_i r D \frac{\partial^2 \psi}{\partial r^2} + N_i r \frac{\partial D}{\partial r} \frac{\partial \psi}{\partial r} + N_i D \frac{\partial \psi}{\partial r} + N_i r D \frac{\partial^2 \psi}{\partial z^2} + N_i r \frac{\partial D}{\partial z} \frac{\partial \psi}{\partial z}] dr dz \quad (B2)$$

Integrating just the second order terms by parts will yield

$$\begin{aligned} \iint_{r,z} N_i D \frac{\partial^2 \psi}{\partial r^2} r dr dz &= \int_z [N_i r D \frac{\partial \psi}{\partial r}]_r dz \\ &\quad - \iint_{r,z} \frac{\partial \psi}{\partial r} [DN_i + rN_i \frac{\partial D}{\partial r} + rD \frac{\partial N_i}{\partial r}] dr dz \quad (B3) \end{aligned}$$

and

$$\iint_{r,z} N_i D \frac{\partial^2 \psi}{\partial z^2} r dr dz = \int_r [N_i D \frac{\partial \psi}{\partial z}]_z r dr - \iint_{r,z} \frac{\partial \psi}{\partial z} [\frac{\partial N_i}{\partial z} D + N_i \frac{\partial D}{\partial z}] r dr dz \quad (B4)$$

Substituting the results of (B3) and (B4) into equation (B2) will give

$$\int_z [N_i r D \frac{\partial \psi}{\partial r}]_r dz + \int_r [N_i D \frac{\partial \psi}{\partial z}]_z r dr - \iint_{r,z} D [\frac{\partial N_i}{\partial r} \frac{\partial \psi}{\partial r} + \frac{\partial N_i}{\partial z} \frac{\partial \psi}{\partial z}] r dr dz \quad (B5)$$

APPENDIX C

LIST OF RELATIONS FOR MATERIAL THERMAL PROPERTIES

A. FUEL (UO_2)

1. Specific Heat, Ref. [19]

$$C_{pF} = [18.45 + 2.431 \times 10^{-3}T - 2.272 \times 10^{-5}T^2] / 270.07$$

[cal/gm °C]

$$T - ^\circ\text{C}$$

2. Thermal Conductivity, Ref. [5]

$$Tk_F = [1 - 2.5(1 - \rho_{TD})] \times \left[\frac{45.1}{135 + T} + 4.79 \times 10^{-13}T^3 \right] \times 0.239$$

[cal/cm sec °C]

$$\rho_{TD} - \text{percent theoretical density}$$

$$T - ^\circ\text{K}$$

B. CLAD (Stainless Steel)

Properties are assumed to be temperature independent, and average values from Ref. [5] were used for the clad properties.

C. COOLANT (Liquid Sodium)

1. Specific Heat, Ref. [5]

$$C_{p_{co}} = 0.34574 - 0.79226 \times 10^{-4}T + 0.34086 \times 10^{-7}T^2$$

[cal/gm °C]

$$T - ^\circ\text{F}$$

2. Density, Ref. [5]

$$\rho_{co} = [59.566 - 7.9504 \times 10^{-3}T - 0.2872 \times 10^{-6}T^2 + 0.06035 \times 10^{-9}T^3] \times 0.01601 \quad [\text{gm/cm}^3]$$

$$T - ^\circ\text{F}$$

3. Thermal Conductivity, Ref. [5]

$$Tk_{co} = [54.306 - 1.878 \times 10^{-2}T + 2.0914 \times 10^{-6}T^2] \\ \times 4.134 \times 10^{-3} \quad [cal/cm \text{ sec } ^\circ C]$$

$$T - ^\circ F$$

D. SURFACE HEAT-TRANSFER COEFFICIENT

$$h_{surf} = \frac{Tk_{co}}{De} [7.0 + 0.025 \left(\frac{De V_{co} \rho_{co} C_{p_{co}}}{Tk_{co}} \right)^{0.8}] \\ [BTU/hr \text{ ft}^2 \text{ } ^\circ F]$$

$$Tk_{co} - [BTU/hr \text{ ft } ^\circ F]$$

$$\rho_{co} - [lbm/ft^3]$$

$$V_{co} - [ft/sec]$$

$$C_{p_{co}} - [BTU/lbm \text{ } ^\circ F]$$

$$De - \text{equivalent diameter [ft]}$$


```

CCMCN/SHAFUN/SN, DESN, DXSN
COMMON/ TIME/ FF, G, RHO
COMMON/ WEIGHT/ WT
FCRMAT (415,F10.5)
10 FCRMAT (0,9X,JSKF = ,I5)
11 FCRMAT (8F10.5)
15 FCRMAT (0,8(8F15.6,/) )
20 FCRMAT (0,10X,ORDER OF GAUSS CLADRATURE IS,I5,/,10X,NUMBER C
25 IF RACIAL FUEL ELEMENTS IS,I5,/,10X,NUMBER CF AXIAL FUEL ELEMENT
30 IS,I5,/,10X,NUMBER OF RADIAL PCINTS IS,I5,/,1CX,
35 FUEL ROD HEIGHT (CM) IS,F10.5)
FCRMAT (F10.5)
3C FCRMAT (0,1, RADIAL NODAL CCCRCINATES',/,5X,13F5.4)
35 FCRMAT (0,14X,DCF,10X,8,11X,E,9X,BETA,8X,BETA1,6X,
3C FCRMAT (0,14X,DCF,10X,8,11X,E,9X,BETA,8X,BETA1,6X,
40 FCRMAT (9X,9E12.5)
41 FCRMAT (0,12X,FFLUX0,8X,VEL,8X,SIGAF,7X,SIGFF,7X,
42 FCRMAT (0,7X,SIGAB,7X,SIGAC,6X,SIGACC)
43 FCRMAT (0,14X,DCC,9X,CPC,8X,DENC,9X,TKC,8X,CCCC,
44 FCRMAT (0,8X,HSURF,8X,DENF)
45 FCRMAT (0,5X,RHC = RHCA + RHCB*T,/,6X,RHCA = ,F10.3,
1 ICX, RPOB = ,F10.3)
INPUT THE ORDER OF THE GAUSS QUADRATURE ---- 3,4, CR 5
INPUT THE NUMBER OF RADIAL ELEMENTS
INPUT THE NUMBER CF AXIAL ELEMENTS
INPUT THE NUMBER CF NODAL PCINTS IN THE RADIAL DIRECTION--2*NFE+7
INPUT THE HEIGHT CF THE FUEL ROD ---- CM
REAL (5,10) NORD,NFE,NEZ,NPR,FEIGHT
WRITE (6,25) NORD,NFE,NEZ,NPR,FEIGHT
INPUT THE RADIAL NODAL DISTANCES
READ (5,30)(DR(I),I=1,NPR)
WRITE (6,35) (DR(I),I=1,NPR)
CALL GINNY (ITYPE,NELCCN,NSTART,R,Z)
CALL CPCOMP (IBP,JA,JB,MAME,NAME,NELCON,NSTART)
CALL INIT (IBP,NELCCN,R,Z,Y)
CALL SHAPE
CC 45 I = 1,JC
BIGF12(I) = 0.0
BIGF3(I) = 0.0
BIGF5(I) = 0.0
BIGK(I) = 0.0
CCNTINUE
DC 50 I = 1,NP
BIGF(I) = C.0
SICRE YIT -- THE INITIAL TEMPERATURE DISTRIBUION
YIT(I) = Y(1,I+NP)

```



```

5C CCNTINUE
CALL MATK1 (NELCCN)
CC 100 MM= 1,NEL
CALL JACOB (MM,NELCCN,R,Z)
CALL MATH12 (MM,BIGH12,BIGH3,BIGF5,BIGF,JA,JB,NAME,NELCCN)
CALL SYSH12 (MM,BIGH12,BIGH3,BIGF5,BIGF,JA,JB,NAME,NELCCN)
IF (NELCCN(11,MM).EQ. C) GO TO 100
CALL SYSK12 (MM,JA,JB,NAME,NELCCN,R,Z,BIGK)
CCNTINUE
1CC INPUT FUEL PHYSICAL CONSTANTS
C *** REAC(5,15) DCF,B,E,BETA,BETAI,DCLAMI,FFLUX0,ANU,VEL,SIGAF,SIGFF,
      SIGAB,SIGFB,RHCA,RHCB,DENF
      AKINF=ANU*SIGFF/SIGAF
C *** INFL CLAD PHYSICAL CONSTANTS
C *** REAC(5,15) DCC,CPC,DENC,TKC,SIGAC
C *** INPUT CCOLANT PHYSICAL CONSTANTS
      READ(5,15) DCCO,SIGACC,VCO,HSURF
      WRITE(6,40)
      WRITE(6,41) DCF,B,E,BETA,BETAI,DCLAMI,ANU,AKINF
      WRITE(6,42)
      WRITE(6,41) FFLUX0,VEL,SIGAF,SIGFF,SIGAB,SIGFB,SIGAC,SIGACC
      WRITE(6,43)
      WRITE(6,41) DCC,CPC,DENC,TKC,DCCC,VCO,HSURF,DENF
      WRITE(6,44) RFOA, RFOB
      JSKF = C
      NY = NP*2
      NL = 0
      N = NY
      EPS = 0.01
      T,TEND,F,HMIN,HMAX
      REAC(5,15) T,TEND,F,HMIN,HMAX,EPS,W,Z,YIT,
      CALL SDESOL(Y,YL,T,TEND,NY,NL,M,JSKF,6,1,F,HMIN,HMAX,EPS,W,Z,YIT,
      BIGH12,BIGH3,BIGH5,BIGK,BIGF,IBP,ITYPE,JA,JB,NAME,NELCCN,R,Z,YIT,
      BIGH4)
      WRITE(6,11) JSKF
      STCP
      ENC
200 1
      2

```



```

SUBROUTINE GINNY (ITYPE,NELCCN,ASTART,R,Z)
SUBROUTINE GINNY DEVELOPES THE SYSTEM MESH
IT ALLOWS FOR ONE OR TWO RADIAL ELEMENTS IN THE FUEL
AND ONE ELEMENT IN THE CLAD AND COCLANT
NFE SETS THE NUMBER OF RADIAL FUEL ELEMENTS
THE GENERATOR ALLOWS FOR ANY NUMBER OF AXIAL ELEMENTS DESIRED
NUMBER OF AXIAL ELEMENTS SET BY NEZ
THE GENERATOR WILL GIVE THE R AND Z DIMENSION FOR EACH NCDE
IT ALSO DEVELOPES THE CONNECTIVE MATRIX

ITYPE -- AN ARRAY USED TO INDICATE THE TYPE OF NCDE
0 -- FUEL NCDE NOT IN AN INTERFACE ELEMENT
1 -- FUEL NCDE IN AN INTERFACE ELEMENT
2 -- CLADDING NODE
3 -- COCLANT NCDE
NELCON -- THE CONNECTIVE MATRIX
R -- RADIAL DIMENSION OF THE ITH NODE
Z -- AXIAL DIMENSION OF THE ITH NCDE

IMPLICIT INTEGER*2(I-N)
DIMENSION DR(13)
DIMENSION ITYPE(1)
DIMENSION NELCON(11,1)
DIMENSION NSTART(1)
DIMENSION R(1), Z(1)
COMMON/CCNN/ NP,NEL,JC
COMMON/INP/DR,HEIGHT,NPR,NEZ,NFE
LL = 0
NP = 1
KK = 1
MM = 2*NFE + 1
MM = M + 4
MMN = M + 2
NEL = NFE + 2
NPFZ = NEZ*2
NACC = 3*NFE + 14
EZ = NEZ
Z1 = HEIGHT/(2.0*EZ)
DO 100 I = 1,NPZ
I = I
IF (KK.EQ. 1) GO TO 80
LL = LL + 1
NSTART(LL) = NP + 1
DO 70 J = 1,NPR
NF = NP + 1
Z(NP) = Z1*T
R(NP) = DR(J)

```

6C


```

70 KK = 1
  CCNTINUE
  CC TO 100 JJ = 1, NPR, 2
  EC IF (JJ.EQ. MMM) GO TO 50
  NF = NP + 1
  Z(NP) = Z1*T
  R(NP) = DR(JJ)
  IF (JJ.EQ. M) .CR. (JJ.EQ. MM)) GO TO 85
  EC TO 9C
  E5 NF = NP + 1
  Z(NP) = Z1*T
  R(NP) = DR(JJ)
  KK = 0
  CCNTINUE
  1CC CCNTINUE
  CC 125 J = 1, NPR
  NP = NP + 1
  Z(NP) = 0
  R(NP) = DR(J)
  125 CCNTINUE
  NNEZ = NEZ - 1
  IF (NNEZ.EQ.0) GO TO 151
  CC 150 LL = 1, NNEZ
  NST = NST + M + 1
  NCLT = NOUT - 2
  NCHA = NOUT + NPR - 3
  NCCUNT = 0
  CC 150 NN = NST, NEND, 2
  IF (NN.EQ. NCHA) GO TO 150
  NN = NN + 1
  NEL = NEL + 1
  NCCUNT = ACCUNT + 1
  IF (NN.EQ. NOUT) NN = NN - 1
  NELCON(1, NEL) = NNN + NADD
  NELCON(2, NEL) = NELCON(1, NEL) - 1
  NELCON(3, NEL) = NELCON(2, NEL) - 1
  NELCON(4, NEL) = NNN + NPR + 1 - ACCUNT
  NELCON(5, NEL) = NNN + 1
  NELCON(6, NEL) = NNN + 2
  NELCON(7, NEL) = NNN + 1
  NELCON(8, NEL) = NELCON(4, NEL) + 1
  CCNTINUE
  15C NST = NST + M + 1
  151 NEND = NST + NPR - 3
  NCLT = NCLT + M + 1
  NCHA = NCLT - 2

```



```

N = 0
153 CC 153 I = AST,NEND,2
II = 0
IF (I.EQ. NOUT) II=1
IF (I.EQ. NCTA) GC TO 153
N = N + 1
NELCCN(1,N) = I + 2 - II
NELCCN(2,N) = I + 1 - II
NELCCN(3,N) = I - II
NELCCN(4,N) = I - NFE - 4 - N - II
IF (I.EQ. NEND) II=2
NELCON(5,N) = NP - NPR + 2*N - 1 + II
NELCCN(6,N) = NELCCN(5,N) + 1
NELCCN(7,N) = NELCCN(5,N) + 2
NELCCN(8,N) = NELCON(4,N) + 1
CCNTINUE
N = NFE + 2
154 DC 155 I = 1,NFE
IF (I.EQ. NFE) GO TO 156
CC 155 J = I,NEL,M
NELCCN(11,J) = 0
CCNTINUE
155 GC TO 159
CC 158 J = I,NEL,M
NELCON(11,J) = 1
NELCCN(11,J+1) = 2
NELCCN(11,J+2) = 3
CCNTINUE
156 CC 180 J = 1,NEL
J = NELCCN(11,J)
CC 180 I = 1,8
II = NELCCN(1,J)
ITYPE(II) = JJ
CCNTINUE
157 CCNTINUE
158 CCNTINUE
159 CC 160 I = 1,NP,3
II = I + 1
IF (II.EQ. NP) GC TO 161
I2 = I + 2
IF (I2.EQ. NP) GO TO 162
WRITE(6,20) I,R(I),Z(I),I1,R(I1),I2,R(I2),Z(I2)
GC TO 165
160 WRITE(6,20) I,R(I), Z(I)
GC TO 165
161 WRITE(6,20) I,R(I), Z(I),I1, R(I1), Z(I1)
162 WRITE(6,25)
163 DC 170 J = 1,NEL

```



```

17C WRITE (6,30) (J, (NELCCN(I,J), I = 1,8))
15 FCFRMT (10X, 'NODAL COORDINATES', //, 10X, 'NCDE', 4X, 'R VALUE', 6X, 'Z V
   1ALUE', 10X, 'NODE', 4X, 'R VALUE', 6X, 'Z VALUE', 4X, 'R VALUE',
   2, 6X, 'Z VALUE', //)
20 FCFRMT (10X, I4, 2(3X, F10.4), 9X, I4, 2(3X, F10.4), 9X, I4, 2(3X, F10.4))
25 FCFRMT (10X, 'CONNECTIVITY MATRIX', //, 10X, 'ELEM', 3X, 'NCD1', 3X, 'NCD2
   1, 3X, 'NCD3', 3X, 'NCD4', 3X, 'NCD5', 3X, 'NCD6', 3X, 'NCD7', 3X, 'NCD8', //)
30 FCFRMT (7X, 5(3X, I4))
   RETURN
END

```



```

SLROUTINE UPCOMP (IBP,JA,JB,NAME,NAME,NELCCN,NSTART)
SLROUTINE CFCOMP CALCULATES THE NAME ARRAY,
THE JA ARRAY, AND THE JB ARRAY WHICH
ARE USED IN THE OPTIMUM COMPACTING SCHEME

IBP -- AN ARRAY USED TO STORE THE FUEL-CLAC INTERFACE NODES
JA -- AN ARRAY WHICH INDICATES THE NUMBER OF NODES CONTRIBUTING
TO THE ITH NODE
JB -- THE POINTER ARRAY WHICH INDICATES WHERE THE ITH
EQUATION BEGINS IN NAME
NAME -- TWC-DIMENSIONAL ARRAY USED TO DEVELOP NAME
NAME -- THE ARRAY USED FOR THE OPTIMUM COMPACTING SCHEME
NELCCN -- CONNECTIVE MATRIX
NSTART -- AN ARRAY USED TO STORE CENTERLINE NCCAL PCINTS

IMPLICIT INTEGER*2(I-N)
DIMENSION DR(13)
DIMENSION IBP(1)
DIMENSION JA(1), JB(1), NAME(1), NAME(26,1)
DIMENSION NELCON(11,1)
DIMENSION NSTART(1)
COMMON/BCUND/ NCCOUNT
COMMON/CCNN/ NP,NEL,JC
COMMON/INP/DR,HEIGHT,NPR,NEZ,NFE
NELCOF = 8
WRITE (6,10) NP,NEL,NELCOF
FCRMTAT(/2X,'NUMNP=',I5,5X,'NUMEL=',I5,5X,'NELDOF=',I5)
10 DC 40 I = 1,NP
JA(I)=1
CCNTINUE
DC 60 I = 1,NP
CC 50 J = 1,26
NAME(J,I)=0
CCNTINUE
CCNTINUE
DC 65 I=1,NP
NAME(1,I)=1
CCNTINUE
DC 100 I = 1,NEL
DC 50 J=1,NELDOF
J2 = NELCCN(J,I)
CC 80 K=1,NELDOF
IF(K.EQ.J) GC TO 80
KK = NELCCN(K,I)
JA(JJ)=JA(JJ)+1
JJA=JA(JJ)

```

CCCCCCCCCCCCCCCC

C


```

CC 7C L=2,JAA
JL=L,MAME(L,JJ)
IF(JJL.EQ.KK) JA(JJ)=JA(JJ)-1
IF(JJL.EQ.KK) GO TO 80
IF(JJL.EQ.0) MAME(JAA,JJ)=KK
CCCONTINUE
7C
EC
50
1CC
C ***
C ***
C ***
TC ACCUNT FOR THE INTERFACE CONCITIONS
ACCUNT = 3
IEP(1) = NP - 6
IBP(2) = NFE + 1
NN = NFE + 7
NLM = 2*NFE
WRITE (6,1000) (NSTART(I),I=1,NEZ)
FCRMAT (10X,I1I5)
DC 300 I = 1,NEZ
IEP(NCOUNT) = NSTART(I) + NUM
IF(I.EQ.NEZ) GC TO 300
IEP(NCOUNT + 1) = IEP(NCOUNT) + NN
ACCUNT = NCGUNT + 2
CCCONTINUE
3CC
WRITE (6,1000) (IBP(I),I=1,ACCUNT)
KK = 1
CC 4C I = 1,ACOUNT
JJ = IBP(I)
IF (KK.EQ.0) GO TO 350
IF (I.EQ.ACOUNT) GO TO 350
IF (I.EQ.1) GO TO 350
CC 325 J = 1,13
MAME (J+13,JJ) = MAME (J,JJ+1)
MAME (J+13,JJ+1) = MAME (J,JJ)
MAME (J+13,JJ+3) = MAME (J,JJ+4)
MAME (J+13,JJ+4) = MAME (J,JJ+3)
CCCONTINUE
325
JA(JJ) = JA(JJ) + 13
JA (JJ+1) = JA(JJ+1) + 13
JA (JJ+3) = JA(JJ+3) + 13
JA (JJ+4) = JA(JJ+4) + 13
KK = 0
GC TO 400
CC 375 J = 1,8
CC 375 J = 1,8
MAME (J+8,JJ) = MAME (J,JJ+1)
MAME (J+8,JJ+1) = MAME (J,JJ)

```



```

375 IF ((I .EQ. 1) .OR. (I .EQ. NEZ1)) JJ = JJ + 1
NAME (J+8, JJ+2) = NAME (J, JJ+3)
NAME (J+8, JJ+3) = NAME (J, JJ+2)
IF ((I .EQ. 1) .OR. (I .EQ. NEZ1)) JJ = JJ - 1
CCNTINUE
JA (JJ) = JA (JJ) + 8
JA (JJ+1) = JA (JJ+1) + 8
IF ((I .EQ. 1) .OR. (I .EQ. NEZ1)) JJ = JJ + 1
JA (JJ+2) = JA (JJ+2) + 8
JA (JJ+3) = JA (JJ+3) + 8
IF (I .EQ. 1) KK = C
CCNTINUE
400
C ***
C ***
JE(I)=1
JC=C
CC2CO I = 1, NP
JN=JA(I)
JE(I+1)=JB(I)+JA(I)
JC=JC+JA(I)
CCNTINUE
20C WRITE(7,5) (JA(I), I=1, NP)
C WRITE(7,5) (JB(I), I=1, NP), JC
FCRMAT(16I5)
FCRMAT(29I4)
205 WRITE(6,2I5) JC
215 FCRMAT(/2X, 'JC=', I10)
DC250 I = 1, NP
JA=JA(I)
JEL=JB(I)
CC 240 J=1, JAA
JJ=JBL+J-1
NAME(JJ)=NAME(J, I)
CCNTINUE
240 CCNTINUE
250 WRITE(7,5) (NAME(I), I=1, JC)
C RETURN
END

```



```

23C A2 = R(NCDES)**2
    R2 = R(NODE)**2
    Y(1,NODE+NP) = Y(1,NCDEM+NP)*(1.0-R2/A2) + Y(1,NODES+NP)*R2/A2
    GC TO 250
    IF((K.EQ. 4).OR.(K.EQ. 8)) GC TC 240
    NCDES = NELCCN(7,M+1)
    NCDEM = NELCCN(5,M)
    A2 = R(NCDES)**2
    R2 = R(NODE)**2
    Y(1,NODE+NP) = Y(1,NODEM+NP)*(1.0-R2/A2) + Y(1,NCDES+NP)*R2/A2
    GC TO 250
    NCDES = NELCCN(8,M+1)
    NCDEM = NELCCN(4,M)
    A2 = R(NCDES)**2
    R2 = R(NODE)**2
    Y(1,NODE+NP) = Y(1,NODEM+NP)*(1.0-R2/A2) + Y(1,NCDES+NP)*R2/A2
    CCA1 INUE
    GC TO 600
    KK = 1
    IF(NFE.EQ. 1) KK = 0
    NCDE = NELCCN(2,M)
    NCDES = NELCCN(1,M)
    NCDEM = NELCCN(3,M-KK)
    A2 = R(NCDE)**2
    R2 = R(NODE)**2
    Y(1,NODE+NP) = Y(1,NODEM+NP)*(1.0-R2/A2) + Y(1,NCDES+NP)*R2/A2
    NCDE = NELCCN(6,M)
    NCDES = NELCCN(7,M)
    NCDEM = NELCCN(5,M-KK)
    Y(1,NODE+NP) = Y(1,NODEM+NP)*(1.0-R2/A2) + Y(1,NCDES+NP)*R2/A2
    GC TC 600
    NCDE1 = NELCCN(1,M) + NP
    NCDE2 = NELCCN(2,M) + NP
    NCDE3 = NELCCN(3,M) + NP
    NCDE4 = NELCCN(4,M) + NP
    NCDE5 = NELCCN(5,M) + NP
    NCDE6 = NELCCN(6,M) + NP
    NCDE7 = NELCCN(7,M) + NP
    NCDE8 = NELCCN(8,M) + NP
    Y(1,NODE1) = Y(1,NCDE3)
    Y(1,NODE2) = Y(1,NCDE3)
    Y(1,NODE6) = Y(1,NCDE5)
    Y(1,NODE7) = Y(1,NCDE5)
    Y(1,NODE8) = Y(1,NCDE4)
    CCA1 INUE
    GC TO 600
    C**
    C**
    C**
    INPUT THE MAXIMUM FLUX (I.E. THE FLUX
    AT THE AXIAL CENTER) RADIAL FLUX WILL BE ASSUMED
    6CC

```



```

C**
C** TC REMAIN CONSTANT (FLAT FLUX ASSUMPTION)
C** AXIALLY THE FLUX ASSUMED TO VARY AS THE
C** SHAPE OF THE SINE FUNCTION
1001 READ(5,1001) FMAX
      FCRMAT (F10.5)
      CC 700 N = 1,NP
      X = Z(M)*3.1415926/HEIGHT
      S = SIN(X)
      Y(1,M) = FMAX*S
700 CONTINUE
      WRITE(6,1050)
1050 FCRMAT (1,9X,'NODAL',5X,'INITIAL',3X,'INITIAL',/,10X,'POINT',5X,
1      'FLUX',6X,'TEMP',//)
      CC 800 I = 1,NP
      WRITE(6,1100) I, Y(1,I),Y(1,I+NP)
800 CONTINUE
1100 FCRMAT (9X,I4,E13.3,F10.3)
      RETURN
      END

```


SLERCUT INE SHAPE

SHAPE EVALUATES THE SHAPE FUNCTION AND
ITS DERIVATIVE WITH RESPECT TO X_1 AND
WITH RESPECT TO ϵ AT EACH OF THE GAUSS POINTS
CONSIDER THE SHAPEALSC SETS THE WEIGHTS AND LOCATION
OF THE GAUSS POINTS DEPENDING UPON THE ORDER
OF GAUSSIAN QUADRATURE USED

IMPLICIT INTEGER*2(I-N)

GP,ETA,SN,DESN,DXSN,XI,ETA1,ETA2,XI1,XI2,ETASC,XISC

REAL*8 XW, WT

CIPENSICN, SN(8,25), DESN(8,25), CXSN(8,25), WT(25), CP(5), XW(5)

COMMON/ GP / GP, NORD

CCMCN/SHAFUN/SN; CESN, CXS

CCMN/WEIGH

2
2
2
2
2

NC - NORD-2
 $\lambda = 0$
 GC TO (4,6,8) NO
 4 GF(1) = 0.7745966692414834

4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 10

[illegible]
$$G_F(\mathbf{r}) = -G_P(\mathbf{r}) = \frac{1}{4\pi} \frac{1}{r^3}$$
$$X_4(1) = 5.0/9.0$$
$$6/0 \cdot 8 = (2)4x$$

0
 11
 (0
 (11
 50
 X0

CC TO 10

$$6 \text{ GF}(1) = 0.861136311594053$$
$$G(2) = 0.339$$
$$G_F(3) = -G_F(2)$$
$$G_F(4) = -G_F(1)$$
$$x_w(1) = 0.347854845137454$$
$$X_h(z) = 0.652$$
$$X_L(E) = X_W(2)$$
$$=$$

CC TO 1C

$$\varepsilon_{\text{CF}}(1) = 0.506179845938664$$
$$GF(2) = 0.5384$$
$$CF(z) = 0.0$$
$$G_F(4) = -G_F(2)$$
$$\begin{pmatrix} 3 \\ 5 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \\ 1 \end{pmatrix} - \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}$$
$$2 \times 10^4 = 0.236926885056189$$
$$X_A(1) = 0.478628670501093$$

X	(1)	0.568
X	(2)	0.568
X	(3)	0.568

$$X_4(4) = X_H(2)$$

()
()
X	X
X	X
X	X
()
()
X	X
X	X
X	X
()
()
X	X
X	X
X	X

```

0  CC 13  I = 1, NORD
    XW(5) = XW(1)

```

二二二二二

UUUUUUUUUU


```

13  W1(N) = XW(I)*XW(J)
    N = 0
    CC 15 I = 1,NORD
    CC 15 GP(I)
    CC 15 J = 1,NORD
    CC 15 GP(J)
    XI = N + 1
    ETAL1 = 1.0 + ETA
    ETAL2 = 1.0 - ETA
    XI1 = 1.0 + XI
    XI2 = 1.0 - XI
    ETASQ = 1.0 - ETA*ETA
    XISC = XI*XI
    CESN(1,N) = 0.25*ETAL1*XI1*(XI+ETA-1.0)
    CESN(1,N) = 0.25*XI1*(2.0*ETA+XI)
    CESN(1,N) = 0.25*ETAL1*(2.0*XI+ETA)
    CESN(2,N) = 0.5*(1.0-XISQ)*ETAL
    CESN(2,N) = SN(2,ETA)/ETAL
    CESN(2,N) = -XI*ETAL1*ETAL1*(-XI+ETA-1.0)
    CESN(3,N) = 0.25*XI2*ETAL1*(2.0*ETA-XI)
    CESN(3,N) = 0.25*XI2*(2.0*XI-ETA)
    CESN(4,N) = C.5*XI2*(1.0-ETASQ)
    CESN(4,N) = -ETAL*XI2
    CESN(4,N) = -SN(4,N)/XI2
    CESN(5,N) = -0.25*XI2*ETAL2*(XI+ETA+1.0)
    CESN(5,N) = 0.25*XI2*(2.0*ETA+XI)
    CESN(5,N) = 0.25*ETAL2*(2.0*XI+ETA)
    CESN(6,N) = 0.5*(1.0-XISQ)*ETAL2
    CESN(6,N) = -SN(6,N)/ETAL2
    CESN(7,N) = -XI*ETAL2*ETAL2*(XI-ETA-1.0)
    CESN(7,N) = 0.25*XI1*(2.0*ETA-XI)
    CESN(7,N) = 0.25*ETAL2*(2.0*XI-ETA)
    CESN(8,N) = 0.5*XI1*(1.0-ETASQ)
    CESN(8,N) = -ETAL*XI1
    CESN(8,N) = SN(8,N)/XI1
    15  RETURN
    END

```



```

C
C
C
C
C
SUBROUTINE MATK1 (NELCCN)
  SUBFCUTINE MATK1 CALCULATES THE 8X8 ELEMENT K1
  AND K2 MATRICES FOR THE ELEMENTS BECARDER IN
  CNE OF THE TWO INTERFACES
  IMPLICIT INTEGER*2 (I-N)
  REAL*8 WT, GP, ETA, XSN{5,5}, XW, Y
  DIMENSION NELCON(11,11)
  DIMENSION XK1(8,8), XK2(8,8)
  COMMON/CCCN/ NP, NEL, JC
  COMMON/CP/ GP, NORD
  COMMON/KIMAT/XK1, XK2
  COMMON /WEIGHT/ WT
  X1 = WT(1)*0.5
  CC 15 I = 1,8
  CC 15 J = 1,8
  XK1(I,J) = 0.0
  XK2(I,J) = 0.0
  CCNTINUE
15 CC 20 K = 1, NORD
  ETA = GP(K)
  XSN(1,K) = 0.5*(ETA+1.0)*ETA
  XSN(2,K) = 0.5*(ETA-1.0)*ETA
  XSN(3,K) = 1.0 - ETA*ETA
  CCNTINUE
30 CC 100 I = 1,3
  CC 80 J = 1,3
  Z = 0.0
  CC 60 K = 1, NORD
  X = XSN(I,K)
  Y = XSN(J,K)
  Z = Z + X*Y*WT(K)/XW
  CCNTINUE
60 XK1(I,J) = Z
  CCNTINUE
80 CCNTINUE
100 XK1(1,7) = XK1(1,2)
  XK1(1,8) = XK1(1,3)
  XK1(7,7) = XK1(2,2)
  XK1(7,8) = XK1(2,3)
  XK1(8,8) = XK1(3,3)
  XK2(3,3) = XK1(1,1)
  XK2(3,4) = XK1(1,3)
  XK2(3,5) = XK1(1,2)
  XK2(4,4) = XK1(3,3)
  XK2(4,5) = XK1(2,3)
120

```



```

SLERCUTINE MATH12 (M,NELCON,R,Z)

SLERCUTINE MATH12 CALCULATES THE 8X8 ELEMENTT H12,
F3 AND H5 MATRICES AND CALCULATES THE 8X1 VECTOR F
F3 IS STORED IN THE H4 ARRAY TO SAVE CORE STORAGE

IMPLICIT INTEGER*2(I-N)
REAL*8 SN,DESN,DXSN,STJ11,STJ22,DETJ,WT,ZX,ZY,ZZ,X,Y,XX,YY,RR
REAL*8 XX,X,YYY,FF,RC,GP
DIMENSION GP(5)
DIMENSION H12(8,8),H3(8,8),F(8)
DIMENSION F4(8,8)
DIMENSION NELCON(11,1)
DIMENSION R(1),Z(1)
DIMENSION SN(8,25),DESN(8,25),WT(25)
DIMENSION STJ11(25),STJ22(25)
COMMON/CONN/ NP,NEL,JC
COMMON/CP/ GP,NORD
COMMON/JACINV/ STJ11,STJ22,DETJ
COMMON/MATRIX/ H12,F3,F
COMMON/MTRXH4/ H4
COMMON/SHAFUN/SN,DESN,DXSN
COMMON/WEIGHT/ WT
NC = NCRD*NORD
LE = NELCON(11,M)
CC 100 I = 1,8
FF = 0.0
CC 80 J = 1,8
ZX = 0.0
ZY = 0.0
ZZ = 0.0
ZZZ = 0.0
CC 60 K = 1,NO
X = DXSN(I,K)*STJ11(K)
XX = DESN(I,K)*STJ11(K)
YY = DESN(J,K)*STJ22(K)
XXY = DESN(I,K)*SN(J,K)
YYY = SN(J,K)
RR = 0.0
CC 40 L = 1,8
LL = NELCON(L,M)
RC = R(LL)
RF = RR + SN(L,K)*RD
CCAT INUE
Z} = ZX+ X*Y*RR*WT(K)
ZY = ZY + XX*YY*RR*WT(K)
ZZ = ZZ + XX*YY*RR*WT(K)

```

4C


```

ZZZ = ZZZ + XXX*YYY*RR*WT(K)
IF(J.NE.1) GO TO 60
IF((LE.EQ.2).OR.(LE.EQ.3)) GO TO 60
FF = FF + XXX*RR*WT(K)
6C CCNTINUE = (ZX + ZZ)*CETJ
F12(I,J) = ZZZ*DETJ
F3(I,J) = ZY*DETJ
F4(I,J) = FF*DETJ
F(I) = FF*DETJ
8C CCNTINUE
ICC CCNTINUE
RETURN
ENC

```



```

CC 45 II = 1,8
LL = NELCCN(II,M)
WRITE(6,888) SN(II,K),Y(1,LL+NP),YIT(LL)
888 FCRMAT(5X,3G18.4,/)
45 CCNTINUE
50 TAU NL = ALOG(TAU1)
60 TT = TAU NL
80 FF = HH + XXX*YYY*RR*WT(K)*TT
90 CCNTINUE = FF*DETJ
EC H4(I,J)
100 CCNTINUE
RETURN
END

```



```

2  JE = NELCCN(3,M)
   CC 220 K = 3,5
   KK = NELCCN(K,M)
   KKK = JA(KK)/2 - 1
   LLL = JB(KK) - 1
   CC 200 I = 3,5
   II = NELCCN(I,M)
   JJ = II - 1
   NN = 0
   NN2 = 0
   CC 180 I = 1, KKK
   LL = LLL + I
   LS = LL + KKK
   KKM = NAME(LL)
   KKS = NAME(LS)
   IF (II .NE. KKM) GC TO 160
   BIGK(LL) = BIGK(LS) + XK2(K,I)
   NN1 = 1
   IF (JJ .NE. KKS) GC TO 170
   BIGK(LS) = BIGK(LS) - XK2(K,I)
   NN2 = 1
   IF (NN1 .EQ. 1) .AND. (NN2 .EQ. 1) GO TO 200
   CC CONTINUE
   CC CONTINUE
   CC CONTINUE
   CC 4 RETURN
   END
160
170
180
200
220

```



```

C
C SLEROUT INE DIFFUN(Y,YL,T,HINV,DY,EIGH12,BIGF3,BIGF4,BIGF5,BIGK
1 EIGF,IBP,ITYPE,JA,JB,NAME,NELCCN,R,Z,VIT)
C
C SLEROUT INE DIFFUN FORMS THE NODAL EQUATIONCS.
C IT SETS THE ITH EQUATION FOR FLUX EQUAL TO CYI
C AND FOR TEMPERATURE TO DYII
C
      IMPLICIT INTEGER*2(I-N)
      REAL*8 CYI,DYII
      DIMENSION BIGH12(1), BIGH3(1), BIGF5(1)
      DIMENSION BIGH4(1)
      DIMENSION BIGK(1)
      DIMENSION BIGF(1)
      DIMENSION YIT(1)
      DIMENSION DR(13)
      DIMENSION IBP(1), ITYPE(1)
      DIMENSION JA(1), JB(1), NAME(1)
      DIMENSION NELCON(11,1)
      DIMENSION R(1), Z(1)
      DIMENSION Y(7,1), CY(1)
      COMMON/RUNC/NCCUNT
      COMMON/CLAD/DCC,CPC,DENC,TKC,SIGAC,HGAP
      COMMON/CCNN/NP,NELJC
      COMMON/CCCL/DCCO,SIGACC,VCO,FSURF
      COMMON/FUEL/DCF,B,E,BETA,BETAI,DCLAMI,FFLUXC,AKINF,VEL,
1 SIGAF1,SIGFF1,SIGAB,SIGFB,DENF
      COMMON/INP/DR,HEIGHT,NPR,NEZ,NFE
      COMMON/TIME/F,G,RHO
      DATA TCLD/-1.0/
      CALCULATE THE TIME DEPENDENT TERMS
      IF (T.EQ.(TOLD)) GO TO 75
      CALL FUNC(T,HINV)
      CALCULATE THE SYSTEM H4 MATRIX
      WFLCH IS TEMPERATURE DEPENDENT
      CC 30 I = 1,JC
      BIGF4(I) = 0.0
      CC 50 I = 1,NEL
      IF (NELCCN(I,I).GT.1) GO TO 50
      CALL MATF4(I,NELCCN,R,Z,VIT,Y)
      CALL SYSF4(I,JA,JB,NAME,NELCON,BIGH4)
      CCONTINUE
      TCLK = T
      VINV = 1.0/VEL
      Z1=Z(IBP(2))-Z(IBP(1))
      EI = 1.0 - BETA
      SIGCFF = SIGFF1
      SIGAF = SIGAF1
      CCNST1 = AKINF*B1 - 1.0

```



```

CCNST2 = SIGAF*B1*RHC
TC = 0.98
FMCD = 1.0/(0.55*FEIGHT)
CC 510 I = 1, NP
JFA = JA(I)
JEB = JB(I)
JAB = JAA + JBB - 1
II = ITYPE(I) + 1
II = NP + I
CYI = 0.0
CYII = 0.0
GC TO (100,200,300,400),IT
100 CC 150 J = JBB, JAB
NN = NN + NP
DVI = DYI + ((DCF*DBLE(BIGH12(J)) - ((CONST1+F)*SIGAF+CCNST2))*
101 DELE(BIGH3(J)) + B*SIGAF*B1*DELE(BIGH4(J)))*Y(1,NN) - G*DBLE(BIGF(NN))
2 VINV*HINV*DBLE(BIGH3(J))*Y(2,NN)
TCK = Y(1,NNN) + 273.
TKF = ((1.0-2.5*(1.0-ID))*(45.1/(135.+TDK) + 4.79E-13*TDK**2))*C.235
CFF = (18.45 + 2.431E-3*TDK - 2.272E5/TDK**2)/270.07
CYII = DYII + TKF*DBLE(BIGH12(J))*Y(1,NNN) + FINV*DENF*CFF*
102 CBLE(BIGH3(J))*Y(2,NNN) - E*SIGFF*CELE(BIGH3(J))*Y(1,NN)
103 CCNTINUE
150 GC TC 500

      DETERMINE THE TYPE CF NCDE BEING CONSIDERED
      K = 0      NCDE NOT CN THE FUEL-CLAD INTERFACE
      K = 1      NCDE CN THE FUEL-CLAD INTERFACE

200 K = 0
DC 210 IB = 1, NCCUNT
IF (IBP(IB) .EQ. I) GO TO 215
210 CCNTINUE
GC TC 220
215 K = 1
220 CC 250 J = JBB, JAB
NN = NAME(J)
NN = NN + NP
DVI = CYI + ((DCF*DBLE(BIGH12(J)) - ((CONST1+F)*SIGAF+CCNST2))*
101 DELE(BIGH3(J)) + B*SIGAF*B1*DELE(BIGH4(J)))*Y(1,NN) - G*DBLE(BIGF(NN))
2 VINV*HINV*DBLE(BIGH3(J))*Y(2,NN)
RA = R(NN)
TCK = Y(1,NNN) + 273.
TKF = ((1.0-2.5*(1.0-ID))*(45.1/(135.+TDK) + 4.79E-13*TDK**2))*C.235
CFF = (18.45 + 2.431E-3*TDK - 2.272E5/TDK**2)/270.07
FMCD = (1000.0 + 247.0*COS(3.14159*(Z(NN)*FMCD-0.81818)))*1.356E-4
CCNST3 = FMCD*Z1

```



```

C
CC
CC
CC
38C
40C
41C
42C
43C
44C
45C
46C
47C
48C
49C
50C
51C
C**
CC**
CC**

CYI = CYI + (DCCO*DELE(BIGH12(JJ)) + SIGACC*DBLE(BIGH3(JJ))) *
1Y(1,NN) + VINV*HINV*DBLE(BIGH3(JJ)) * Y(2,NN)
CCCONTINUE
IF(K.NE. 2) GO TO 500
CYI = Y(1,I) - Y(1,I-1)
GC TO 500

DETERMINE THE TYPE CF NODE BEING CONSIDERED
K = 1
NODE NOT ON CLAD-COOLANT INTERFACE
K = 2
NODE ON THE CLAD-COOLANT INTERFACE
K = 1
NCC = 0
DC 420 IB = 1, NCCOUNT
LEF = IEP(1B) + 4
IF(NODC.EQ. 0) GO TO 410
LEF = LBP - 1
NCC = C
GC TO 415
NCCD = 1
IF(LEP.EQ. 1) GO TO 425
CCCONTINUE
GC TO 430
K = 2
DC 480 J = JBB, JAB
NA = NAME(J)
NNA = NN + NP
IF(K.EC. 2) GO TO 450
CYI = DYI + (DCCO*DELE(BIGH12(J)) + SIGACC*DELE(BIGH3(J))) *
1Y(1,NN) + VINV*HINV*DBLE(BIGH3(J)) * Y(2,NN)
45C RE = R(NN)
TCF = 1.8 * Y(1,NN) + 32.0
TKCO = (54.306 - 1.878E-2 * TCF + 2.0914E-6 * TCF**2) * 4.134E-3
CENCC = (59.566 - 7.9504E-3 * TDF - 0.2872E-6 * TCF**2 +
1C.06035E-9 * TDF**3) * C.016C1
CFCC = 0.34574 - 0.79226E-4 * TDF + C.34086E-7 * TDF**2
CYII = CYII + (TKCC*DBLE(BIGH12(J)) + RB*HSURF*Z1*CELE(BICK(J)) +
1VCC*CECCO*CPCCO*DBLE(BIGH5(J))) * Y(1,NN) + CENCC*CFCC*
2CELE(BIGH3(J)) * HINV*Y(2,NN)
CCCONTINUE
IF(K.NE. 2) GO TO 500
CYI = Y(1,I) - Y(1,I-1)
CY(I) = DYI
CY(II) = DYII
CCCONTINUE
51C
C**
CC**
CC**

TC ACCOUNT FOR THE BOUNDARY CONDITIONS

```


C**

```
NP = NP + 1
NN = NPF*2
NN = NPF - NN
CC 600 I = NN,NP
CY(I) = 0.0
CCCONTINUE
```

600

C**

C**

C

C

C**

C**

TC ACCOUNT FOR THE TEMP BOUNDARY CONDITIONS
(CONSTANT COOLANT INLET TEMPERATURES)

```
NP = NP - 2
CC 700 I = NN,NP
CY(I+NP) = 0.0
CCCONTINUE
CCRETURN
CCEND
```

700


```

1  + B*SIGAF*B1*BIGH4(J)
TCK = Y(1,NNN) + 273.
TKF = ((1.0-2.5*(1.0-ID)))*(45.1/(135.+TDK) + 4.79E-13*(TDK**2)))*C.235
CPF = (18.45 + 2.431E-3*TDK - 2.272E5/TDK**2)/270.07
PW(J+JC) = TKF*BIGH12(J)+DENF*CPF*AH*BIGH3(J)
CCNTINUE
GC TC 500

15C
CETERMINE THE TYPE CF NODE BEING CCNSIDERED
K = 0 NCDE NOT CN THE FUEL-CLAD INTERFACE
K = 1 NCDE CN THE FUEL-CLAD INTERFACE
K = C
CC 210 IB = 1, NCCUNT
IF(1BP(IB) .EQ. I) GO TC 215
CCNTINUE
GC TC 220
20C
K = 1
CC 250 J = JBB, JAB
PF(J) = DCF*BIGH12(J) - ((CONST1+F)*SIGAF+CONST2-AH*VINV)*BIGH3(J)
1 + B*SIGAF*B1*BIGH4(J)
NN = NAME(J) + NF
RA = R(NN)
FGAP = (1000.0 + 247.0*COS(3.14159*(Z(NN)*+MCD-0.81818)))*1.356E-4
CCNST3 = HGAP*Z1
TCK = Y(1,NNN) + 273.
TKF = ((1.0-2.5*(1.0-ID)))*(45.1/(135.+TDK) + 4.79E-13*(TDK**2)))*C.235
CPF = (18.45 + 2.431E-3*TDK - 2.272E5/TDK**2)/270.07
PW(J+JC) = TKF*BIGH12(J)+DENF*CPF*AH*BIGH3(J)+RA*CCNST3*BIGHK(J)
IF(K .EQ. 0) GO TC 250
IF(J .LT. JAB + J - JAB) GO TO 250
PW(J) = DCC*BIGH12(JJ) + (SIGAC + AH*VINV)*BIGH3(JJ)
IF(J .NE. JAB) GO TO 25C
PW(JBB) = PW(JBB) + PW(J)
PW(J) = 0.0
CCNTINUE
GC TO 500

25C
CETERMINE THE TYPE CF NODE BEING CCNSIDERED
K = 1 NCDE NOT CN AN INTERFACE
K = 2 NCDE CN THE FUEL-CLAD INTERFACE
K = 3 NCDE ON THE CLAD-COOLANT INTERFACE
K = 1
NCC = 0
CC 320 IB = 1, NCCUNT
LEF = 1BP(IB) + 1
30C

```



```

410 GC TO 415
411 NCCC = 1
412 IF(LBP.EG. 1) GO TO 425
420 CCNTINUE
421 GC TC 430
422 K = 2
423 CC 48C J = JBB,JAB
424 NN = NAME(J)
425 NN = NAME(J) + NF
426 PW(J) = 0.0
427 IF(K.EG. 2) GO TO 450
428 PW(J) = DCCO*BIGH12(J) + (SIGACC + AH*VINV)*BIGH3(J)
429 RE = R(NN)
430 TCF = 1.8*Y(1,NNN) + 32.0
431 TKCO = (54.306 - 1.878E-2*TCF + 2.0914E-6*TCF**2)*4.134E-3
432 CPCC = C.34574 - 0.79226E-4*TCF + C.34086E-7*TCF**2
433 DCCCC = (59.566 - 7.9504E-3*TCF - 0.2872E-6*TCF**2 +
1 C.06C35E-9*TCF**3)*C.016C1
434 A = DCCCC*PCO
435 PW(J+JC) = TKCO*BIGH12(J) + VCO*A*BIGH5(J) + AT*A*BIGHF3(J)
436 I + RB*HSURF*Z1*BIGHK(J)
437 CCNTINUE
438 IF(K.NE. 2) GO TC 500
439 PW(JBB) = 1.0
440 PW(JA3) = -1.0
441 CCNTINUE
442 C***
443 C***
444 C***
445 C***
446 C***
447 C***
448 C***
449 C***
450 C***
451 C***
452 C***
453 C***
454 C***
455 C***
456 C***
457 C***
458 C***
459 C***
460 C***
461 C***
462 C***
463 C***
464 C***
465 C***
466 C***
467 C***
468 C***
469 C***
470 C***
471 C***
472 C***
473 C***
474 C***
475 C***
476 C***
477 C***
478 C***
479 C***
480 C***
481 C***
482 C***
483 C***
484 C***
485 C***
486 C***
487 C***
488 C***
489 C***
490 C***
491 C***
492 C***
493 C***
494 C***
495 C***
496 C***
497 C***
498 C***
499 C***
500 C***
501 C***
502 C***
503 C***
504 C***
505 C***
506 C***
507 C***
508 C***
509 C***
510 C***
511 C***
512 C***
513 C***
514 C***
515 C***
516 C***
517 C***
518 C***
519 C***
520 C***
521 C***
522 C***
523 C***
524 C***
525 C***
526 C***
527 C***
528 C***
529 C***
530 C***
531 C***
532 C***
533 C***
534 C***
535 C***
536 C***
537 C***
538 C***
539 C***
540 C***
541 C***
542 C***
543 C***
544 C***
545 C***
546 C***
547 C***
548 C***
549 C***
550 C***
551 C***
552 C***
553 C***
554 C***
555 C***
556 C***
557 C***
558 C***
559 C***
560 C***
561 C***
562 C***
563 C***
564 C***
565 C***
566 C***
567 C***
568 C***
569 C***
570 C***
571 C***
572 C***
573 C***
574 C***
575 C***
576 C***
577 C***
578 C***
579 C***
580 C***
581 C***
582 C***
583 C***
584 C***
585 C***
586 C***
587 C***
588 C***
589 C***
590 C***
591 C***
592 C***
593 C***
594 C***
595 C***
596 C***
597 C***
598 C***
599 C***
600 C***
601 C***
602 C***
603 C***
604 C***
605 C***
606 C***
607 C***
608 C***
609 C***
610 C***
611 C***
612 C***
613 C***
614 C***
615 C***
616 C***
617 C***
618 C***
619 C***
620 C***
621 C***
622 C***
623 C***
624 C***
625 C***
626 C***
627 C***
628 C***
629 C***
630 C***
631 C***
632 C***
633 C***
634 C***
635 C***
636 C***
637 C***
638 C***
639 C***
640 C***
641 C***
642 C***
643 C***
644 C***
645 C***
646 C***
647 C***
648 C***
649 C***
650 C***
651 C***
652 C***
653 C***
654 C***
655 C***
656 C***
657 C***
658 C***
659 C***
660 C***
661 C***
662 C***
663 C***
664 C***
665 C***
666 C***
667 C***
668 C***
669 C***
670 C***
671 C***
672 C***
673 C***
674 C***
675 C***
676 C***
677 C***
678 C***
679 C***
680 C***
681 C***
682 C***
683 C***
684 C***
685 C***
686 C***
687 C***
688 C***
689 C***
690 C***
691 C***
692 C***
693 C***
694 C***
695 C***
696 C***
697 C***
698 C***
699 C***
700 C***
701 C***
702 C***
703 C***
704 C***
705 C***
706 C***
707 C***
708 C***
709 C***
710 C***
711 C***
712 C***
713 C***
714 C***
715 C***
716 C***
717 C***
718 C***
719 C***
720 C***
721 C***
722 C***
723 C***
724 C***
725 C***
726 C***
727 C***
728 C***
729 C***
730 C***
731 C***
732 C***
733 C***
734 C***
735 C***
736 C***
737 C***
738 C***
739 C***
740 C***
741 C***
742 C***
743 C***
744 C***
745 C***
746 C***
747 C***
748 C***
749 C***
750 C***
751 C***
752 C***
753 C***
754 C***
755 C***
756 C***
757 C***
758 C***
759 C***
760 C***
761 C***
762 C***
763 C***
764 C***
765 C***
766 C***
767 C***
768 C***
769 C***
770 C***
771 C***
772 C***
773 C***
774 C***
775 C***
776 C***
777 C***
778 C***
779 C***
780 C***
781 C***
782 C***
783 C***
784 C***
785 C***
786 C***
787 C***
788 C***
789 C***
790 C***
791 C***
792 C***
793 C***
794 C***
795 C***
796 C***
797 C***
798 C***
799 C***
800 C***
801 C***
802 C***
803 C***
804 C***
805 C***
806 C***
807 C***
808 C***
809 C***
810 C***
811 C***
812 C***
813 C***
814 C***
815 C***
816 C***
817 C***
818 C***
819 C***
820 C***
821 C***
822 C***
823 C***
824 C***
825 C***
826 C***
827 C***
828 C***
829 C***
830 C***
831 C***
832 C***
833 C***
834 C***
835 C***
836 C***
837 C***
838 C***
839 C***
840 C***
841 C***
842 C***
843 C***
844 C***
845 C***
846 C***
847 C***
848 C***
849 C***
850 C***
851 C***
852 C***
853 C***
854 C***
855 C***
856 C***
857 C***
858 C***
859 C***
860 C***
861 C***
862 C***
863 C***
864 C***
865 C***
866 C***
867 C***
868 C***
869 C***
870 C***
871 C***
872 C***
873 C***
874 C***
875 C***
876 C***
877 C***
878 C***
879 C***
880 C***
881 C***
882 C***
883 C***
884 C***
885 C***
886 C***
887 C***
888 C***
889 C***
890 C***
891 C***
892 C***
893 C***
894 C***
895 C***
896 C***
897 C***
898 C***
899 C***
900 C***
901 C***
902 C***
903 C***
904 C***
905 C***
906 C***
907 C***
908 C***
909 C***
910 C***
911 C***
912 C***
913 C***
914 C***
915 C***
916 C***
917 C***
918 C***
919 C***
920 C***
921 C***
922 C***
923 C***
924 C***
925 C***
926 C***
927 C***
928 C***
929 C***
930 C***
931 C***
932 C***
933 C***
934 C***
935 C***
936 C***
937 C***
938 C***
939 C***
940 C***
941 C***
942 C***
943 C***
944 C***
945 C***
946 C***
947 C***
948 C***
949 C***
950 C***
951 C***
952 C***
953 C***
954 C***
955 C***
956 C***
957 C***
958 C***
959 C***
960 C***
961 C***
962 C***
963 C***
964 C***
965 C***
966 C***
967 C***
968 C***
969 C***
970 C***
971 C***
972 C***
973 C***
974 C***
975 C***
976 C***
977 C***
978 C***
979 C***
980 C***
981 C***
982 C***
983 C***
984 C***
985 C***
986 C***
987 C***
988 C***
989 C***
990 C***
991 C***
992 C***
993 C***
994 C***
995 C***
996 C***
997 C***
998 C***
999 C***
1000 C***

```



```

CC 700 I = NN,NP
FW(JB(I)+JC) = 1.0
JS = JB(I) + 1 + JC
JE = JB(I) + JA(I) - 1 + JC
CC 700 J = JS,JE
FW(J) = 0.0
CCCONTINUE
RETURN
END
700

```



```

C
C
C
C
      SLROUTINE NUTISL (PW,DY,F1,N,NY,EPS,YMAX,NEWPW,KRET,BIGH3,
1  ITYPE,JA,JB,NAME,NELCCN)
      SLROUTINE NUTISL CALCULATES THE NEWTON-ITERATES
      USING A 'SOR' TECHNIQUE WITH AN OVER-RELAXATION
      FACTOR EQUAL TO OMEGA1
      IMPLICIT INTEGER*2(I-N)
      INTEGER*4 N,NY,NEWPW,KRET
      REAL*8 PN,TN,BIGH3(1)
      DIMENSION ITYPE(1)
      DIMENSION JA(1),JB(1),NAME(1)
      DIMENSION NELCON(11,1)
      DIMENSION PW(1),DY(1),F1(1),YMAX(1)
      COMMON/CCNN/ NP,NEL,JC
      COMMON/CFUEL/CCF,B,E,BETA,BETAI,DCLAMI,FFLUXC,AKINF,VEL,
1  SIGAF,SIGFF,SIGAB,SIGFB,DENF
      DATA CMEG,OMEGM1/1.02,0.02/
      TEST = 1.E35
      KRET = 0
      EPS = EPS*2
      EPSSA2 = EPSS*0.0001
      NCIT = N
      E1 = E*SIGFF
      DO 100 I = 1,NP
      F1(I) = DY(I)/PW(JB(I))
      F1(I+NP) = DY(I+NP)/PW(JB(I)+JC)
      DO 300 ITT = 1,NCIT
      RCF = 0.
      CF = 0.
      RCF1 = 0.
      RCF2 = 0.
      CF1 = 0.
      CF2 = 0.
      DO 200 I = 1,NP
      JS = JB(I) + 1
      JE = JB(I+1) - 1
      ITYPE(I)
      PN = DY(I)
      TN = DY(I+NP)
      CC 150 J = JS
      PN = PN - DBLE(PW(J))*F1(NAME(J))
      TN = TN - DBLE(PW(J+JC))*F1(NAME(J)+NP)
      IF(IT.GT.1) GO TO 150
      TN = TN + E1*DBLE(BIGH3(J))*F1(NAME(J))
      CC 15C CCNTINUE
      PN = PN/PW(JS-1)

```



```

TN = TN/PW(JS+JC-1)
FN = FN*OMEG - F1(I)*OMEGM1
TN = TN*OMEG - F1(I+NP)*OMEGM1
PCH = F1(I) - PN
TCH = F1(I+NP) - TN
IF ((ABS(PCH).GT.TEST).OR.(ABS(TCH).GT.TEST))GC TC 18C
CF1 = CH1 + (PCH/YMAX(I))*2
CF2 = CH2 + (TCH/YMAX(I+NP))*2
RCH1 = RCH1 + (PCH/AMAX1(ABS(SNGL(PN)),EPS))*2
RCH2 = RCH2 + (TCH/AMAX1(ABS(SNGL(TN)),EPS))*2
GC TO 190
18C CF1 = 1.
    CF2 = 1.
    RCH1 = 1.
    RCH2 = 1.
    FN = 1.
    TN = 1.
    PN = 1.
    CH1 = 1.
    CH2 = 1.
    RCH1 = 1.
    RCH2 = 1.
    EPSS = 1.
    EPSA2 = 1.
    CCNT = 1.
    KRET = 1.
    RETURN
    END
19C F1(I) = 1.
20C F1(I+NP) = 1.
    CH1 = 1.
    CH2 = 1.
    RCH1 = 1.
    RCH2 = 1.
    EPSS = 1.
    EPSA2 = 1.
    CCNT = 1.
    KRET = 1.
    RETURN
    END
30C

```



```

DIMENSION Y(7,1), YL(1), SAVE(7,1), YMAX(1), ER(1), YLSV(1), FL(1) LDA
1 PERT(6,3), COF(21), ES(1), DY(1), PW(1), SAV(1), A(25) LDA
2 EQUIVALENCE (A(8),END), (A(9),BR), (A(10),E), (A(11),EDWN), LDA
3 (A(12),ENQ1), (A(13),ENQ2), (A(14),ENQ3), (A(15),EPS), (A(16),EUP) LDA
4 (A(17),HNEW), (A(18),PEPSH), (A(19),IDCUB), (A(20),IWEVAL), LDA
5 (A(21),K), (A(22),LCOPYL), (A(23),LCOPYY), (A(24),MAXCER), LDA
6 (A(25),M1), (A(26),NL), (A(27),NC), (A(28),NS), (A(29),NW) LDA
----- LDA
THE CCEFFICIENTS IN THE PERT ARRAY ARE USED FOR ERROR TESTING AND
CHANGING STEP SIZE AND NEED TO BE ACCURATE TO ONLY A FEW DIGITS.
----- LDA
DATA PERT/4.,9.,16.,25.,36.,49.,5.,16.,25.,36.,49.,64.,1.,1.,.25, LDA
12.7889E-2,1.70569E-3,6.83929E-5/ LDA
----- LDA
THE ENTRIES IN THE COF ARRAY ARE THE COEFFICIENTS FOR THE STIFFLY
STABLE METHODS USED IN THIS PROGRAM AND ARE TO BE THE MACHINE
PRECISION EQUIVALENTS OF THE FOLLOWING CONSTANTS.
----- LDA
-1/2, -1/2 -1/6 -1/6 -1/24 -1/24 -1/120 -1/120
-11/6, -1 -35/24, -5/12, -17/24, -1/8 -35/144, -7/240, -1/720
-25/12, -15/8, -17/24, -49/48, -203/90, -49/48, -35/144, -7/240, -1/720
-127/60, -15/8, -17/24, -49/48, -203/90, -49/48, -35/144, -7/240, -1/720
-147/60, -203/90, -49/48, -203/90, -49/48, -35/144, -7/240, -1/720
----- LDA
DATA COF/-1.,-1.5,-.5,-1.833333,-1.,-.1666667,-2.083333,-1.458333, LDA
1-.4166667,-.04166667,-2.283333,-1.875,-.7083333,-.125,-.08333333, LDA
2-2.45,-2.255556,-1.020833,-.2430556,-.02516667,-.00138889, LDA
IF (JSTART) 100,110,150 LDA
----- LDA
IF THIS IS A RESTART ENTRY, RESTORE Y AND YL FROM THE SAVE AND
YLSV ARRAYS, WHERE THEY WERE SAVED BY A PREVIOUS CALL TO LCASAV.
----- LDA
100 CALL COPYZ (Y,SAVE,LCOPYY) LDA
CALL COPYZ (YL,YLSV,LCOPYL) LDA
GC TC 150 LDA
----- LDA
IF THIS IS THE FIRST CALL, INITIALIZE YMAX, SCALE DERIVATIVES, AND
INITIALIZE INDICATORS AND SET ORDER TO ONE.
FOR DOUBLE PRECISION, SET LCOPYL = 14*NY AND LCOPYL = 2*NL IF
SLE ROUTINE COPYZ IS IN SINGLE PRECISION.
----- LDA
110 NL = N-NY LDA

```

CCCCCCCC

CCCCCCCCCCCCCCCC

CCCC

CCCCCCCC


```

LCCPYY = 7*NY
LCCPYL = NL
MI = MINO(M,NY)
EFS = SQR(T(FLOAT(MI))*RMSEPS
MAXCER = MINO(MAXOR,6)
IF (IPRT.LE.0) GO TO 120
PRINT 3, N,NL,RMSEPS,TEND,H
12C NS = C
NW = 0
C
CC 130 J=1,NY
YMAX(J) = AMAX1(1.,ABS(Y(1,J)))
13C Y(2,J) = Y(2,J)*H
C
NG = 1
BR = 1.
ASSIGN 190 TO IRET
C
SET COEFFICIENTS FOR THE ORDER CURRENTLY BEING USED.
EUP IS A TEST FOR ERRORS OF THE CURRENT ORDER NG
EUP IS TO TEST FOR INCREASING THE ORDER, EDWN FOR DECREASING THE
ORDER.
14C K = NG*(NG-1)/2
CALL COPYZ (A(2),COF(K+1),NG)
K = NG+1
ICUB = NG
ENG1 = .5/NG
ENG2 = .5/K
ENG3 = .5/(NG+2)
PEPST = EPS*#2
EUP = PERT(NG,1)*PEPST
EDWN = PERT(NG,2)*PEPSH
BND = (EPS*ENG3)*#2
IWEVAL = 1
CC TO IRET, (190,200,49C,570)
15C IF (H.EC.HNEW) GO TO 190
C
IF CALLER HAS CHANGED I, RESCALE DERIVATIVES TO REFLECT THAT I NEW
WAS USED ON THE LAST CALL.
C
R = H/HNEW
ASSIGN 190 TO IRET
CC TO 610
C
SET JSTART TO NQ, THE CURRENT ORDER OF THE METHCC, BEFCRE EXIT,

```



```

C      AND SAVE THE CURRENT STEPSIZE IN FNEW.
C      160 JSTART = NC
C      FNEW = F
C      RETURN
C      170 NS = NS+1
C      IF (IPRT.LE.0) GO TO 180
C      PRINT DATA IF DESIRED BY USER
C      PRINT 1, NS,NW,NQ,F,T,(Y(1,I),I=1,NY)
C      180 CC CONTINUE
C      IF (KFLAG.LT.0) GO TO 160
C      IF (T.GE.TEND) GO TO 160
C      TAKE ANOTHER STEP IF T < TEND
C      JSTART = 1
C      SAVE DATA FOR TRIAL WITH A SMALLER TIMESTEP IF THIS STEP FAILS
C      190 CALL CCOPYZ (SAVE,Y,LCCOPY)
C      CALL CCOPYZ (YLSV,YL,LCCPYL)
C      RACUM = 1.
C      KFLAG = 1
C      FCLD = F
C      NCCLC = NQ
C      TCCLC = T
C      T = T+H
C      200 FINV = 1./F
C      COMPUTE PREDICTED VALUES BY EFFECTIVELY MULTIPLYING DERIVATIVE
C      VECTOR BY PASCAL TRIANGLE MATRIX
C      CC 210 J=2,K
C      J3 = K+J-1
C      CC 210 J1=J,K
C      J2 = J3-J1
C      CC 210 I=1,NY
C      210 Y(J2,I) = Y(J2,I)+Y(J2+1,I)
C      CC 220 I=1,NY
C      220 ER(I) = 0.

```

```

LDA 2300
LDA 2310
LDA 2320
LDA 2330
LDA 2340
LDA 2350
LDA 2360
LDA 2370
LDA 2380
LDA 2390
LDA 2400
LDA 2420
LDA 2430
LDA 2440
LDA 2450
LDA 2460
LDA 2470
LDA 2480
LDA 2490
LDA 2500
LDA 2510
LDA 2520
LDA 2530
LDA 2540
LDA 2550
LDA 2560
LDA 2580
LDA 2590
LDA 2600
LDA 2610
LDA 2620
LDA 2630
LDA 2640
LDA 2650
LDA 2660
LDA 2670
LDA 2680
LDA 2690
LDA 2700
LDA 2710
LDA 2720
LDA 2730
LDA 2740
LDA 2750
LDA 2760
LDA 2770
LDA 2780

```



```

C CC      CC UP TC THREE CORRECTOR ITERATIONS. CONVERGENCE IS OBTAINED WHEN
C CC      CHANGES ARE LESS THAN BND WHICH IS DEPENDENT ON THE ERROR TEST
C CC      CONSTANT. THE SUM OF CORRECTIONS IS ACCUMULATED IN ER(I). IT IS
C CC      EQUAL TO THE K-TH DERIVATIVE OF Y TIMES F**K/(K-FACTCRIAL*(A(K))),
C CC      AND THUS IS PROPORTIONAL TO THE ACTUAL ERRORS TO THE LOWEST POWER
C CC      CF F PRESENT, WHICH IS F**K.
C CC      -----
C CC      CC 270 L=1,3
C CC      CALL DIFFUN(Y,YL,T,HINV,DY,BIGH12,BIGH3,BIGH4,BIGH5,BIGF,BIGF,
C CC      I JEP,I TYPE,JA,JB,NAME,NELCON,RR,Z,YIT)
C CC      IF (IWEVAL.LT.1) GC TC 230
C CC      -----
C CC      IF THERE HAS BEEN A CHANGE OF ORDER OR THERE HAS BEEN TROUBLE
C CC      WITH CONVERGENCE, PW IS RE-EVALUATED PRIOR TO STARTING THE
C CC      CORRECTOR ITERATION. IWEVAL IS THEN SET TO -1 AS AN INDICATOR
C CC      THAT IT HAS BEEN DONE. NEWPW IS SET NONZERO TO INDICATE TO
C CC      SUBROUTINE NUTSL THAT A NEW PW HAS BEEN PROVIDED.
C CC      -----
C CC      CALL JACMAT(Y,YL,T,HINV,A(2),N,NY,EPS,DY,F1,PW,BIGH12,BIGH3,
C CC      I BIGH4,BIGH5,BIGF,IBF,I TYPE,JA,JB,NAME,NELCON,RR,Z)
C CC      KFLAG = 1
C CC      IWEVAL = -1
C CC      NA = NW+1
C CC      NEWPW = 1
C CC      CALL NUTSL(PW,DY,F1,N,NY,EPS,YMAX,NEWPW,KRET,EIGH3,I TYPE,
230 C CC      JA,JB,NAME,NELCON)
C CC      IF (KRET.NE.O) GO TO 600
C CC      IF (NL.LE.O) GO TO 250
C CC      -----
C CC      CC 240 I=1,NL
C CC      YL(I) = YL(I)-F1(I+NY)
C CC      -----
C CC      CC CONTINUE
C CC      DEL = 0.
C CC      -----
C CC      CC 260 I=1,NY
C CC      Y(1,I) = Y(1,I)-F1(I)
C CC      Y(2,I) = Y(2,I)+A(2)*F1(I)
C CC      ER(I) = ER(I)+F1(I)
C CC      CEL = DEL+(F1(I)/AMAX1(YMAX(I),ABS(Y(1,I))))**2
C CC      CC CONTINUE
C CC      -----
C CC      IF (L.GE.2) BR = AMAX1(.9*BR,DEL/CEL)
C CC      DEL = DEL
C CC      IF (AMIN1(DEL,BR*DEL*.5).LE.BND) GO TO 330
C CC      CC CONTINUE
270 C CC

```



```

C      THE CORRECTOR ITERATION FAILED TO CONVERGE IN 3 TRIES.  VARICUS
C      POSSIBILITIES ARE CHECKED FOR.  IF H IS ALREADY HAIN AND PW PAS
C      ALREADY BEEN RE-EVALUATED, A NO CONVERGENCE EXIT IS TAKEN.
C      OTHERWISE THE MATRIX FW IS RE-EVALUATED AND/CR (IN THAT CR CER) THE
C      STEP IS REDUCED TO TRY AND GET CONVERGENCE.
C      -----
C      T = TOLD
C      IF (IWEVAL) 280,300,290
C      IF (H.LE.HMIN*1.00001) GO TO 310
C      RACUM = RACUM*.25
C      CC CONTINUE
C      GC TO 560
C      KFLAG = -3
C      -----
C      RESTORE Y AND YL AFTER CONVERGENCE FAILURE
C      -----
C      CALL CCPYZ (Y,SAVE,LCCPPY)
C      CALL COPYZ (YL,YLSV,LCCPYL)
C      F = HOLD
C      NC = NQCCLC
C      GC TO 170
C      -----
C      THE CORRECTOR CONVERGED, SO NOW THE ERROR TEST IS MADE.
C      -----
C      C = 0.
C      CC 340 I=1,M1
C      YN = AMAX1((ABS(Y(1,I)),YMAX(I))
C      C = D+(ER(I)/YM)*.2
C      IWEVAL = 0
C      IF (D.GT.E) GO TO 280
C      -----
C      THE ERROR TEST IS CKAY, SO THE STEP IS ACCEPTED.  IF IDOUB
C      NOW BECOMES NEGATIVE, A TEST IS MADE TO SEE IF THE STEP SIZE
C      CAN BE INCREASED AT THIS ORDER OR ONE HIGHER OR ONE LOWER.
C      THE CHANGE IS MADE ONLY IF THE STEP CAN BE INCREASED BY AT
C      LEAST 10%.  IDOUB IS SET TO NQ TO PREVENT FURTHER TESTING
C      FOR A WHILE.  IF NC CHANGE IS MADE, IDOUB IS SET TO 5.
C      -----
C      IF (K.LT.3) GO TO 360
C      CC 250 J=3,K
C      CC 350 I=1,NY
C      CC Y(J,I) = Y(J,I)+A(J)*ER(I)

```



```

C 36C KFLAG = 1 IDCUB-1
      ICCUB = IDCUB-1
      IF (IDCUB) 410,37C,510
      CALL CCPYZ (ESV,ER,M1)
      GC TO 510
      LDA 372C
      LDA 373C
      LDA 374C
      LDA 375C
      LDA 376C
      LDA 377C
      LDA 378C
      LDA 379C
      LDA 380C
      LDA 381C
      LDA 382C
      LDA 383C
      LDA 384C
      LDA 385C
      LDA 386C
      LDA 387C
      LDA 388C
      LDA 389C
      LDA 390C
      LDA 391C
      LDA 392C
      LDA 393C
      LDA 394C
      LDA 395C
      LDA 396C
      LDA 397C
      LDA 398C
      LDA 399C
      LDA 400C
      LDA 401C
      LDA 402C
      LDA 403C
      LDA 404C
      LDA 405C
      LDA 406C
      LDA 407C
      LDA 408C
      LDA 409C
      LDA 410C
      LDA 411C
      LDA 412C
      LDA 413C
      LDA 414C
      LDA 415C
      LDA 416C
      LDA 417C
      LDA 418C
      LDA 419C

-----
C 37C THE ERROR TEST FAILED. IF JSTART = 0, THE DERIVATIVES IN THE
      SAVE ARRAY ARE UPDATED. TESTS ARE THEN MADE TO FIX THE SIZE
      AND PERHAPS REDUCE THE ORDER. AFTER RESTORING AND SCALING THE
      Y VARIABLES, THE STEP IS RETRIED.
      LDA 372C
      LDA 373C
      LDA 374C
      LDA 375C
      LDA 376C
      LDA 377C
      LDA 378C
      LDA 379C
      LDA 380C
      LDA 381C
      LDA 382C
      LDA 383C
      LDA 384C
      LDA 385C
      LDA 386C
      LDA 387C
      LDA 388C
      LDA 389C
      LDA 390C
      LDA 391C
      LDA 392C
      LDA 393C
      LDA 394C
      LDA 395C
      LDA 396C
      LDA 397C
      LDA 398C
      LDA 399C
      LDA 400C
      LDA 401C
      LDA 402C
      LDA 403C
      LDA 404C
      LDA 405C
      LDA 406C
      LDA 407C
      LDA 408C
      LDA 409C
      LDA 410C
      LDA 411C
      LDA 412C
      LDA 413C
      LDA 414C
      LDA 415C
      LDA 416C
      LDA 417C
      LDA 418C
      LDA 419C

C 38C IF (JSTART.GT.0) GC TO 400
      DC 390 I=1,NY
      39C SAVE(2,I) = Y(2,I)
      LDA 372C
      LDA 373C
      LDA 374C
      LDA 375C
      LDA 376C
      LDA 377C
      LDA 378C
      LDA 379C
      LDA 380C
      LDA 381C
      LDA 382C
      LDA 383C
      LDA 384C
      LDA 385C
      LDA 386C
      LDA 387C
      LDA 388C
      LDA 389C
      LDA 390C
      LDA 391C
      LDA 392C
      LDA 393C
      LDA 394C
      LDA 395C
      LDA 396C
      LDA 397C
      LDA 398C
      LDA 399C
      LDA 400C
      LDA 401C
      LDA 402C
      LDA 403C
      LDA 404C
      LDA 405C
      LDA 406C
      LDA 407C
      LDA 408C
      LDA 409C
      LDA 410C
      LDA 411C
      LDA 412C
      LDA 413C
      LDA 414C
      LDA 415C
      LDA 416C
      LDA 417C
      LDA 418C
      LDA 419C

C 40C KFLAG = KFLAG-2
      IF (F.LE.FMIN) GC TO 550
      I = TOLC
      IF (KFLAG.LE.-5) GC TO 530
      41C PR2 = (C/E)**ENQ2*1.2
      L = 0
      IF (NQ.LE.1) GC TO 430
      D = 0.
      LDA 372C
      LDA 373C
      LDA 374C
      LDA 375C
      LDA 376C
      LDA 377C
      LDA 378C
      LDA 379C
      LDA 380C
      LDA 381C
      LDA 382C
      LDA 383C
      LDA 384C
      LDA 385C
      LDA 386C
      LDA 387C
      LDA 388C
      LDA 389C
      LDA 390C
      LDA 391C
      LDA 392C
      LDA 393C
      LDA 394C
      LDA 395C
      LDA 396C
      LDA 397C
      LDA 398C
      LDA 399C
      LDA 400C
      LDA 401C
      LDA 402C
      LDA 403C
      LDA 404C
      LDA 405C
      LDA 406C
      LDA 407C
      LDA 408C
      LDA 409C
      LDA 410C
      LDA 411C
      LDA 412C
      LDA 413C
      LDA 414C
      LDA 415C
      LDA 416C
      LDA 417C
      LDA 418C
      LDA 419C

C 42C DC 420 J=1,M1
      YM = AMAX1(ABS(Y(1,J)),YMAX(J))
      C = C+(Y(K,J)/YM)**2
      LDA 372C
      LDA 373C
      LDA 374C
      LDA 375C
      LDA 376C
      LDA 377C
      LDA 378C
      LDA 379C
      LDA 380C
      LDA 381C
      LDA 382C
      LDA 383C
      LDA 384C
      LDA 385C
      LDA 386C
      LDA 387C
      LDA 388C
      LDA 389C
      LDA 390C
      LDA 391C
      LDA 392C
      LDA 393C
      LDA 394C
      LDA 395C
      LDA 396C
      LDA 397C
      LDA 398C
      LDA 399C
      LDA 400C
      LDA 401C
      LDA 402C
      LDA 403C
      LDA 404C
      LDA 405C
      LDA 406C
      LDA 407C
      LDA 408C
      LDA 409C
      LDA 410C
      LDA 411C
      LDA 412C
      LDA 413C
      LDA 414C
      LDA 415C
      LDA 416C
      LDA 417C
      LDA 418C
      LDA 419C

C 43C PR1 = (C/EDWN)**ENC1*1.3
      IF (PR1.GE.PR2) GC TO 430
      PR2 = PR1
      L = -1
      IF (KFLAG.LT.0.OR.NQ.GE.MAXDER) GC TO 450
      C = 0
      LDA 372C
      LDA 373C
      LDA 374C
      LDA 375C
      LDA 376C
      LDA 377C
      LDA 378C
      LDA 379C
      LDA 380C
      LDA 381C
      LDA 382C
      LDA 383C
      LDA 384C
      LDA 385C
      LDA 386C
      LDA 387C
      LDA 388C
      LDA 389C
      LDA 390C
      LDA 391C
      LDA 392C
      LDA 393C
      LDA 394C
      LDA 395C
      LDA 396C
      LDA 397C
      LDA 398C
      LDA 399C
      LDA 400C
      LDA 401C
      LDA 402C
      LDA 403C
      LDA 404C
      LDA 405C
      LDA 406C
      LDA 407C
      LDA 408C
      LDA 409C
      LDA 410C
      LDA 411C
      LDA 412C
      LDA 413C
      LDA 414C
      LDA 415C
      LDA 416C
      LDA 417C
      LDA 418C
      LDA 419C

C 44C DC 440 J=1,M1
      YM = AMAX1(ABS(Y(1,J)),YMAX(J))
      C = C+(Y(K,J)/YM)**2
      LDA 372C
      LDA 373C
      LDA 374C
      LDA 375C
      LDA 376C
      LDA 377C
      LDA 378C
      LDA 379C
      LDA 380C
      LDA 381C
      LDA 382C
      LDA 383C
      LDA 384C
      LDA 385C
      LDA 386C
      LDA 387C
      LDA 388C
      LDA 389C
      LDA 390C
      LDA 391C
      LDA 392C
      LDA 393C
      LDA 394C
      LDA 395C
      LDA 396C
      LDA 397C
      LDA 398C
      LDA 399C
      LDA 400C
      LDA 401C
      LDA 402C
      LDA 403C
      LDA 404C
      LDA 405C
      LDA 406C
      LDA 407C
      LDA 408C
      LDA 409C
      LDA 410C
      LDA 411C
      LDA 412C
      LDA 413C
      LDA 414C
      LDA 415C
      LDA 416C
      LDA 417C
      LDA 418C
      LDA 419C

C 45C PR1 = (C/ELP)**ENC3*1.4
      IF (PR1.GE.PR2) GC TO 450
      PR2 = PR1
      L = 1
      R = 1./AMAX1(PR2,1.E-5)
      IF (KFLAG.LT.0.OR.R.GE.1.1) GC TO 460
      ICCUB = 9
      LDA 372C
      LDA 373C
      LDA 374C
      LDA 375C
      LDA 376C
      LDA 377C
      LDA 378C
      LDA 379C
      LDA 380C
      LDA 381C
      LDA 382C
      LDA 383C
      LDA 384C
      LDA 385C
      LDA 386C
      LDA 387C
      LDA 388C
      LDA 389C
      LDA 390C
      LDA 391C
      LDA 392C
      LDA 393C
      LDA 394C
      LDA 395C
      LDA 396C
      LDA 397C
      LDA 398C
      LDA 399C
      LDA 400C
      LDA 401C
      LDA 402C
      LDA 403C
      LDA 404C
      LDA 405C
      LDA 406C
      LDA 407C
      LDA 408C
      LDA 409C
      LDA 410C
      LDA 411C
      LDA 412C
      LDA 413C
      LDA 414C
      LDA 415C
      LDA 416C
      LDA 417C
      LDA 418C
      LDA 419C

```



```

460 GC TC 510
    NEWQ = NQ+L
    K = NEWQ+1
    IF (NEWQ.LE.NQ) GC TO 480
    R1 = A(NEWQ)/FLOAT(NEWQ)
C
470 DC 470 J=1,NY
    Y(K,J) = ER(J)*R1
C
480 CC CONTINUE
C
C IF THE STEP WAS OKAY, SCALE THE Y VARIABLES IN ACCORDANCE
C WITH THE NEW VALUE OF F. IF KFLAG < 0, HOWEVER, USE THE
C SAVED VALUES (IN SAVE AND YLSV). IN EITHER CASE, IF THE ORDER
C HAS CHANGED IT IS NECESSARY TO FIX CERTAIN PARAMETERS BY CALLING
C THE PROGRAM SEGMENT AT STATEMENT NUMBER 140.
C
490 IDCLB = NQ
    IF (NEWQ.EQ.NQ) GC TO 490
    NC = NEWQ
    ASSIGN 490 TO IRET
    GC TO 140
495 IF (KFLAG.GT.0) GC TO 500
    RACUM = RACUM*R
    GC TO 560
500 R = AMAX1(AMIN1(HMAX/H,R),HMIN/H)
    F = H*R
    I EVAL = 1
    ASSIGN 510 TO IRET
    GC TO 610
C
510 DC 520 I=1,M1
520 YMAX(I) = AMAX1(ABS(Y(1,I)),YMAX(I))
C
C GC TO 170
C
C THE ERROR TEST HAS NOW FAILED THREE TIMES, SO THE DERIVATIVES ARE
C IN BAD SHAPE. RETURN TO FIRST ORDER METHOD AND TRY AGAIN. CF
C COURSE, IF NQ = 1 ALREADY, THEN THERE IS NO PCPE AND WE EXIT WITH
C KFLAG = -4.
C
530 IF (NQ.EQ.1) GO TC 540
    NC = 1
    ICCUB = 1
    ASSIGN 570 TO IRET
    GC TO 140
540 NCCLD = 1
    KFLAG = -4

```



```

TC RESTART THE USER FIRST CALLS LDCARST TO RESTORE THE VALUES SAVED
BY LDCASAV, THEN RE-ENTERS LDASUB WITH JSTART < 0, AND WITH THE
OTHER PARAMETERS THE SAME AS RETURNED FROM THE LAST ENTRY TO
LDASUB, PARTICULARLY THOSE ARRAYS MENTIONED ABOVE.
-----
ENTRY LDCASAV(SAV)
LCCPYS = 25
CALL COPYZ (SAV,A,LCCPYS)
CALL COPYZ (SAVE,Y,LCCPYV)
CALL COPYZ (YLSV,YL,LCCPYL)
RETURN
ENTRY LDCARST(SAV)
LCCPYR = 25
CALL COPYZ (A,SAV,LCCPYR)
RETURN
-----
1 FCRMAT (2I5,I2,1P2E10.2,7E14.6/(32X,7E14.6))
2 FCRMAT (32X,1P7E14.6)
3 FCRMAT (I1,N=I,I3,NL=I,I3, RMSEPS=I,IPES.2, TENC=I
1 ,ES.2, H=I,ES.2//) H=I,8X,I T I,8X,IY(I,*) AND YL(*)'//)
4 FCRMAT (I NS NW Q
END

```

CCCCC

C

CCCC


```

SUBROUTINE SDESOL (Y, YL, T, TEND, NY, NL, M, JSKF, MAXDER, IPRT, F, HMIN,
1 FMAX, RMSEPS, W, BIGH12, BIGH3, BIGH5, BIGK, BIGF, IBP, ITYPE, JA, JB, NAME,
2 NELCON, R, Z, YIT, BIGH4)
-----
SUBROUTINE SDESOL IS A DRIVER ROUTINE FOR SUBROUTINE LDASUB.
ITS PURPOSE IS TO SET UP THE NECESSARY REFERENCES TO A LARGE
BLOCK OF AUXILIARY STORAGE, AND OBTAIN INITIAL VALUES OF
DERIVATIVES.
THE CALLING SEQUENCE FOR SDESOL IS

CALL SDESOL(Y, YL, T, TEND, NY, NL, M, JSKF, MAXDER, IPRT, F, HMIN, FMAX, RMSEPS, W)
WHERE THE PARAMETERS ARE DEFINED AS FOLLOWS.

Y      - ARRAY DIMENSIONED (7, NY). THIS ARRAY CONTAINS THE
        DEPENDENT VARIABLES AND THEIR SCALED DERIVATIVES.
        Y(J+1, I) CONTAINS THE J-TH DERIVATIVE OF THE I-TH VARIABLE.
        IABLE TIMES H*J/J-FACTORIAL, WHERE H IS THE CURRENT
        STEP SIZE. ON FIRST ENTRY THE CALLER SUPPLIES THE
        INITIAL VALUES OF EACH VARIABLE IN Y(1, I). ON SUB-
        SEQUENT ENTRIES IT IS ASSUMED THE ARRAY HAS NOT
        BEEN CHANGED. TO INTERPOLATE TO NON-MESH POINTS,
        THESE VALUES CAN BE USED AS FOLLOWS. IF H IS THE
        CURRENT STEP SIZE AND VALUES AT TIME T+E ARE
        NEEDED, LET S = E/H AND THEN

                JS      SUM Y(J+1, I)*S**J
                I-TH VARIABLE AT T+E IS J=0

THE VALUE OF JS IS OBTAINED IN THE CALLING PROGRAM
BY JS = IABS(JSKF/10)
ARRAY OF NL VARIABLES WHICH APPEAR LINEARLY.
- CURRENT VALUE OF THE INDEPENDENT VARIABLE (TIME)
- END TIME
- NUMBER OF DIFFERENTIAL EQUATIONS AND NONLINEAR
  VARIABLES.
- NUMBER OF LINEAR VARIABLES
- NUMBER OF VARIABLES INCLUDED IN THE ERROR TEST
- AN INDICATOR USED BOTH ON INPUT AND OUTPUT
  ON INPUT, JSKF = -1 INDICATES A RESTART CALL TO
  SDESOL. JSKF = 0 INDICATES AN INITIAL CALL TO
  SDESOL. JSKF > 0 INDICATES A CONTINUATION OF THE
  PREVIOUS CALL TO SDESOL. JSKF < -1 MAY HAVE RESULTS
  FROM THE USER NEGLECTING TO TEST FOR ERROR RETURNS
  FROM SDESOL. BECAUSE OF THIS POSSIBILITY, JSKF < -1

```



```

RESULTS IN TERMINATION OF THE RUN WITH THE
APPROPRIATE COMMENT.  CONSISTS OF TWO DIGITS AND SIGN,
ON OUTPUT, JSKF CONSISTS OF THE FORMULA CURRENTLY
+ OR - CF.  Q IS THE ORDER OF THE TYPE OF RETURN, AS
BEING USED.  P INDICATES THE TYPE OF RETURN, AS
FOLLOWS.
JSKF > 0, P = 1 IS THE NORMAL RETURN
JSKF < 0, P IS AN ERROR RETURN, WITH THE FOLLOWING
MEANINGS.
P = 1 ERROR TEST FAILURE FOR F > FMIN
P = 3 CORRECTOR FAILED TO CONVERGE FOR F > FMIN
P = 4 CORRECTOR FAILED TO CONVERGE FOR FIRST
ORDER METHOD
P = 5 ERROR RETURN FROM SUBROUTINE NLITSL
P = 6 ERROR RETURN FROM SUBROUTINE DERSL
MAXCER - MAXIMUM ORDER DERIVATIVE THAT SHOULD BE USED IN
METHOD.  IT MUST BE NO GREATER THAN SIX.
IPRT - INTERNAL PRINT CONTROL INDICATOR FOR LCASUB.
IPRT = 0 NO PRINT COUNTERS, STEP SIZE, CURRENT TIME
AND VALUES OF DEPENDENT VARIABLES AT
EACH STEP.
F - CURRENT STEP SIZE.  AN INITIAL VALUE MUST BE SUPPLIED
BUT NEED NOT BE THE ONE WHICH MUST BE USED, SINCE THE
SUBROUTINE WILL CHOOSE A SMALLER ONE IF NECESSARY TO
KEEP THE ERROR PER STEP SMALLER THAN THE SPECIFIED
VALUE.  IT IS BETTER TO UNDERESTIMATE THE INITIAL
STEP SIZE THAN TO OVERESTIMATE IT.  THE STEP SIZE IS
NORMALLY NOT CHANGED BY THE USER.
FMIN - MINIMUM STEP SIZE ALLOWED
HMAX - MAXIMUM STEP SIZE ALLOWED
RMSEPS - THE ERROR TEST CONSTANT.  THE ROOT-MEAN-SQUARE OF
THE SINGLE STEP ERROR ESTIMATES, DIVIDED BY
YMAX(I) = (MAXIMUM STEP SIZE AND/OR THE ORDER
LESS THAN EPS.  TO ACHIEVE THIS,
ARE VARIED TO OBTAIN THIS.
W - SCRATCH STORAGE ARRAY.  MUST BE AT LEAST 13*NY + 5*NL
LOCATIONS, PLUS THE DESCRIPTION OF THE
MATRIX PW (SEE DESCRIPTION OF SUBROUTINE JACMAT).
THE STORAGE OF PW WILL NORMALLY REQUIRE MORE THAN
N*2 + 2*N LOCATIONS, AND IF COMPACT
TECHNIQUES ARE USED, CAN BE MUCH FEWER.
-----
INTEGER*2 IBP, ITYPE, JA, JB, NAME, NELCCN
DIMENSION Y(7,1), YL(1), W(1)
DIMENSION BIGH12(1), BIGH3(1), BIGH5(1)
DIMENSION BIGH4(1)
-----

```



```

C DIMENSION BIGK(1)
C DIMENSION BIGF(1)
C DIMENSION IBP(1), ITYPE(1), NAME(1)
C DIMENSION JA(1), JB(1), Z(1)
C DIMENSION NELCON(1,1)
C DIMENSION R(1), YIT(1)
C DIMENSION YIT(1)
IF (JSKF.GT.0) GO TO 12C
IF (JSKF.LT.-1) GO TO 14C
N = NY+NL
IF (JSKF.LT.0) GO TO 11C

IF THIS IS THE FIRST ENTRY, OBTAIN VALUES OF THE DERIVATIVES.
CALL DERVAL(Y,YL,T,N,NY,W,KRETR,BIGH12,BIGF3,BIGH4,BIGF5,PICK,
1 EIGF,IBP,ITYPE,JA,JB,NAME,NELCCN,R,Z,YIT)
IF (KRETR.NE.0) GC TO 13C

NCH SET UP STORAGE BLOCKS IN THE W ARRAY. THIS NEEDS TO BE DONE
ONLY INITIALLY AND CN RESTARTS.

THE ARRAY SAVE STARTS AT LCCATION 1 NSVL
THE ARRAY YLSV STARTS AT LCCATION NSVL
THE ARRAY YMAX STARTS AT LCCATION NYMAX
THE ARRAY ER STARTS AT LCCATION NERV
THE ARRAY ESV STARTS AT LCCATION NESV
THE ARRAY F1 STARTS AT LCCATION NFI
THE ARRAY DY STARTS AT LCCATION NCI
THE MATRIX PW STARTS AT LCCATION NFW

11C NSVL = 7*NY+1
    NYMAX = NSVL+NL
    NERV = NYMAX+NY
    NESV = NERV+NY
    NFI = NESV+NY
    NCI = NFI+N
    NFW = NCI+N
    JS = JSKF

12C CALL LDASUB(Y,YL,T,TEND,N,NY,M,JS,KF,MAXDEF,IPRT,F,FMIN,FMAX,
    IRMSEPS,W(W(NSVL)),W(NYMAX),W(NERV),W(NESV),W(NFI),W(NDY),W(NPW)),
    1 BIGH12,BIGH3,BIGH4,BIGH5,BIGF,BIGF,IBP,ITYPE,JA,JB,NAME,NELCCN,
    3R,Z,YIT)

CCCE JSKF ON RETURN FROM LDASUB

JSKF = ISIGN(JS*10+IABS(KF),KF)
RETURN
JSKF = -6
RETURN

13C

```



```
14C PRINT 1, JSKF  
STOP
```

```
C  
C
```

```
1 FCFORMAT ('OIT IS AN ERRGR TO ENTER SDESOL WITH JSKF = ', I10//  
1, RUN HAS BEEN TERMINATED.')  
END
```

```
SDE 131C  
SDE 1320  
SDE 1330  
SDE 1340  
SDE 1350  
SDE 1360  
SDE 1370
```


C	SUBROUTINE COPYZ(S,Y,L)	COP	10
C	DIMENSION S(1),Y(1)	COP	20
C		COP	30
C		COP	40
C		COP	50
		COP	60
		COP	70
		COP	80
		COP	90
		COP	100
		COP	110
		COP	120


```

THIS SUBROUTINE COPIES THE ARRAY Y, OF LENGTH L, INTO THE ARRAY S
-----
IF(L.LE.0)RETURN
DO 100 J=1,L
  S(J) = Y(J)
100 RETURN
END

```



```

C      DIMENSION Y(7,1), YL(1), W(1)
C      DC 100 I=1,NY
C      W(2*N+1) = AMAX1(ABS(Y(1,I)),1.)
C      100 Y(3,I) = 0.
C
C      HINV = 16.*20
C      KERET = 0
C      EPS2 = NY/1.E5
C      EPS = SCRT(EPS2)
C
C      CC 140 IT=1,10
C
C      DC 110 I=1,NY
C      110 Y(2,I) = Y(3,I)/HINV
C
C      CALL DIFFUN (Y,YL,T,FINV,W,BIGH12,BIGH3,BIGH4,BIGH5,BIGK,BIGF,
C      1 IBP,ITYPE,JA,JB,NAME,NELCON,R,Z,YIT)
C      CALL JACMAT (Y,YL,T,FINV,-1.,NY,NY,EPS,W,W(N+1),W(3*N+1),BIGH12,
C      1 BIGH3,BIGH4,BIGH5,BIGK,IBP,ITYPE,JA,JB,NAME,NELCCN,R,Z)
C      NEWPW = 1
C
C      CC 120 I=1,NY
C      120 W(1) = W(1)*HINV
C
C      CALL NUTSL (W(3*N+1),W,W(N+1),NY,NY,EPS,W(2*N+1),NEWPW,KRET,
C      1 BIGH3,ITYPE,JA,JB,NAME,NELCCN)
C      IF (KRET.NE.0) GC 1C 170
C      ER = 0.
C
C      CC 130 I=1,NY
C      Y(3,I) = Y(3,I)-W(N+1)
C      W(2*N+1) = AMAX1(ABS(Y(3,I)),1.)
C      130 ER = ER+(W(N+1)/AMAX1(ABS(Y(3,I)),1.))**2
C
C      IF (ER.LT.EPS2) GC TO 150
C      CC CONTINUE
C      GC TO 170
C
C      CC 150 CC 160 I=1,NY
C      160 Y(2,I) = Y(3,I)
C
C      RETURN
C      KERET = 1
C      170 RETURN
C      END

```


LIST OF REFERENCES

1. Apostolakis, G. E., "Analytical Estimate of the Error in Conventional Point-Kinetic Reactivity Due to Spatial Effects," Nucl. Sci. Engr., 53, p. 141-152, 1974.
2. Yasinsky, J. B., "On the Use of Point Kinetics for the Analysis of Rod Ejection Accidents," Nucl. Sci. Engr., 39, p. 241-256, 1970.
3. Jackson, J. F. and Kastenburg, W. S., "Space-time Dynamics Studies in Large LMFBR's with Feedback," Nucl. Sci. Engr., 42, p. 278-294, 1970.
4. Nguyen, D. H. and D. Salinas, "Finite Element Solution of Space-Time Nonlinear Reactor Dynamics," Nucl. Sci. Engr., 60, p. 120-130, 1976.
5. Hanford Engineering Development Laboratory Report HEDL-TME 92-128, User's Guide for GAPCON: A Computer Program to Predict Fuel-to-Cladding Heat Transfer Coefficients in Oxide Fuel Pins, by G. R. Horn and F. E. Panisko, 1972.
6. Hanford Engineering Development Laboratory Report HEDL-TME 74-47, Melt III-A Neutronics Thermal-Hydraulic Computer Program for Fast Reactor Safety, v. 1, by A. S. Walter and others, 1974.
7. Hanford Engineering Development Laboratory Report HEDL-TME 75-50, An Analysis of the Unprotected Transient Overpower Accident in the FTR, by A. S. Walter and others, 1975.
8. Naval Postgraduate School Report NPS-69Zc-76111, An Optimal Compact Storage Scheme for Nonlinear Reactor Problems by FEM, by D. Salinas, D. H. Nguyen, and R. Franke, 1976.
9. Naval Postgraduate School Report NPS-53Fe-76051, A Program for the Numerical Solution of Large Sparse Systems of Algebraic and Implicitly Defined Stiff Differential Equations, by R. Franke, 1976.
10. Lamarsh, J. R., Introduction to Nuclear Reactor Theory, Addison, Wesley, 1972.
11. Salinas, D., D. H. Nguyen and T. W. Southworth, "Finite Element Solution of a Nonlinear Nuclear Reactor Dynamics Problem," International Conference on Computational Methods in Nonlinear Mechanics, Austin, Texas, p. 541-550, 1974.

12. Semenze, L. A., Lewis, E. E., and Rossow, E. C., "The Application of the Finite Element Method to the Multi-group Neutron Diffusion Equation," Nucl. Sci. Engr., 47, p. 302-310, 1972.
13. Zienkiewicz, O. C., The Finite Element Method in Engineering Science, McGraw-Hill, 1970.
14. Zlamal, M., "On the Finite Element Method," Numer. Meth., 12, p. 394-409, 1968.
15. Personal conversation with Dr. G. Cantin of the Naval Postgraduate School, 1976.
16. Cook, R. D., Concepts and Applications of Finite Element Analysis, Wiley, 1974.
17. Zienkiewicz, O. C. and C. J. Parekh, "Transient Field Problems: Two-Dimensional and Three-Dimensional Analysis by Isoparametric Finite Elements," Int. J. for Numer. Meth. in Engr., v. 2, p. 61-71, 1970.
18. Barsoum, R. S., "On the Use of Isoparametric Finite Elements in Linear Fracture Mechanics," Int. J. for Numer. Meth. in Engr., v. 10, no. 1, p. 25-37, 1976.
19. Belle, J., Uranium Dioxide: Properties and Nuclear Applications, Naval Reactors, Div. of Reactor Development, U. S. Atomic Energy Commission, p. 189, 1961.
20. Lew, G. T., A Three Dimensional Solution of the Transient Field Problem Using Isoparametric Finite Elements, Masters Thesis, Naval Postgraduate School, 1972.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 69 Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
4. Dr. D. H. Nguyen (thesis advisor) 4625 Larchmont NE Albuquerque, New Mexico 87115	1
5. Assoc. Professor D. Salinas (thesis advisor) Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
6. Assoc. Professor R. Franke (second reader) Department of Mathematics Naval Postgraduate School Monterey, California 93940	1
7. Professor Gilles Cantin Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
8. LT R. E. Kasdorf, USN (student) 129 Redondo Ct. Marina, California 93933	2

7 FEB 78

24834

168562

Thesis
K14495
c 1

Kasdorf

Finite element solu-
tion of the nonlinear
coupled neutronic-
energy equations for a
fast reactor fuel cell.

7 FEB 78

24834

Thesis
K14495
c.1

Kasdorf

Finite element solu-
tion of the nonlinear
coupled neutronic-
energy equations for a
fast reactor fuel cell.

168562

thesK14495

Finite element solution of the nonlinear



3 2768 002 11448 0

DUDLEY KNOX LIBRARY